

Begonnen am	Mittwoch, 9. April 2025, 13:58
Status	Beendet
Beendet am	Mittwoch, 9. April 2025, 15:57
Verbrauchte Zeit	1 Stunde 58 Minuten
Punkte	27,43/28,00
Bewertung	36,24 von 37,00 (97,96%)

Frage 1

Richtig

Erreichte Punkte 5,00 von 5,00

v1 (neueste)

Geben Sie die NF ein, die der Frage entspricht:

Es gibt keine transitiven Abhängigkeiten von Schlüsseln

NF3



Attributwerte müssen atomar sein

NF1



Wenn sich eine Tabelle in NF2 befindet, ist sie automatisch in ...

NF1



Nicht-Schlüssel-Attribute sind voll funktional abhängig von Schlüsseln

NF2



Prüfen Sie jedes Attribut, das nicht zu den Schlüsselkandidaten gehört, ob es von allen Schlüsselkandidaten oder nur einem Teil davon abhängig ist.

NF2



Es gibt kein Nicht-Schlüsselattribut, das nur von einem Teil des Schlüssels abhängt

NF2



Abhängigkeiten zwischen Nicht-Schlüsselattributen prüfen

NF3



Aufteilen von Attributen mit mehreren Werten

NF1



Abhängigkeiten von Nicht-Schlüsselattributen von Schlüsselattributen prüfen

NF2



Wenn Attribute atomar sind und der Schlüsselkandidat nur aus einem Attribut besteht, befindet sich die Tabelle automatisch in ...

NF2



Überprüft die Werte der Attribute

NF1



... prüft auf implizite Abhängigkeiten

NF3



Frage 2

Richtig

Erreichte Punkte 4,00 von 4,00

v1 (neueste)

Vergleiche die Tabelle mit NF1 und beantworte anschließend die gestellten Fragen (keine Attribute hinzufügen oder löschen)

C_no	Surname	First_name	Sex	City	Street	Phone
1	Burton	James	M	Vienna	Höchstädtplatz	01/1234567
2	Falk	Andy	M	Vienna	Herrengasse	01/33333
3	Huber	Toby	M	Linz	Bahnhofstr.	0664/1234
4	Brown	Peter	M	Graz	Neutorgasse	0650/11111
5	Jackson	Anna	F	Vienna	Ring	01/33333 0664/83278
6	Jackson	Tom	M	Vienna	Ring	01/33333
7	Maier	Sandra	F	Graz	Herrengasse	

Richtige Antworten markieren:

Wählen Sie eine oder mehrere Antworten:

- ☒ c_no --> surname, first_name ✓
- ☐ c_no kann als Schlüsselkandidat verwendet werden
- ☒ (c_no, phone) kann als Primärschlüssel verwendet werden ✓
- ☒ C_no --> surname, first_name, sex, city, street ✓
- ☐ city --> street
- ☐ (surname, first_name, city, street) kann als Primärschlüssel verwendet werden
- ☐ surname, first_name, sex, city, street, phone --> c_no
- ☐ phone --> c_no, surname, first_name, sex, city, street
- ☐ (surname, first_name, city, street) kann als Schlüsselkandidat verwendet werden
- ☐ Tabelle ist bereits in NF1
- ☐ c_no kann als Primärschlüssel verwendet werden
- ☐ wenn die Tabelle in NF1 ist, ist die Tabelle dann in NF2?
- ☐ surname --> firstname

Mark correct answers:

Frage 3

Richtig

Erreichte Punkte 10,00 von 10,00

v1 (neueste)

Betrachten Sie das folgende Beispiel:

1. In einer Hochschule gibt es Lektor, die sich genauer durch Vor- und Nachnamen auszeichnen.
2. In dieser Hochschule gibt es verschiedene Fächer (=Studienrichtungen), die sich durch einen eindeutigen Code und eine Beschreibung auszeichnen.
3. An jedem Fach sind ein oder mehrere Lektoren beteiligt, ein einzelner Lektor unterrichtet in einem oder mehreren Fächern.
4. Für jedes Fach gibt es einen Fachreferenten, der auch Lektor ist. Ein Lektor kann Experte für ein, mehrere oder kein Fach sein.

Wenn Sie alle Attribute in einer Tabelle zusammenfassen, haben Sie Folgendes:

college (lect_ID, first_name, surname, subject, sub_description, is_speaker)

Machen Sie Folgendes:

- Finden Sie im ersten Schritt alle funktionalen Abhängigkeiten
- Entscheiden Sie dann, welche(s) Attribut(e) Primärschlüssel sein können.
- Entwickeln Sie die Normalformen (beginnend mit NF1, dann NF2 und zu guter Letzt NF3)

Wenn Sie all dies getan haben, beantworten Sie die folgenden Fragen:

Wählen Sie eine oder mehrere Antworten:

- ☐ lect_ID --> first_name, surname, subject, sub_description, is_speaker
- ☒ lect_id --> first_name ✓
- ☒ lect_id --> surname ✓
- ☐ lect_id --> is_speaker
- ☒ lect_id --> first_name, surname ✓
- ☒ subject --> sub_description, is_speaker ✓
- ☒ lect_id, subject --> first_name, surname, sub_description, is_speaker ✓
- ☒ surname --> surname ✓
- ☒ first_name, surname --> lect_ID (stimmt nicht, da es möglicherweise unterschiedliche Lektoren mit dem gleichen Nachnamen und Vornamen gibt) ✓
- ☒ lect_ID --> is_speaker (stimmt nicht, denn ein Lektor kann Referent für mehrere Fächer sein) ✓
- ☒ subject --> is_speaker (stimmt, denn der Sprecher eines Faches hängt nur vom Fach und nicht vom Lektor ab) ✓
- ☒ subject --> is_speaker (ist wahr, weil der Sprecher eines Themas für ein Thema einzigartig ist) ✓
- ☐ lect_id kann als Primärschlüssel verwendet werden
- ☐ subject kann als Primärschlüssel verwendet werden
- ☒ lect_ID und subject können zusammen als Primärschlüssel verwendet werden ✓
- ☐ is_speaker muss Teil des Primärschlüssels sein
- ☐ first_name, surname, subject kann als Primärschlüssel verwendet werden
- ☐ die angegebene Tabelle befindet sich in NF2, da der Primärschlüssel nur ein Attribut enthält
- ☒ die angegebene Tabelle muss auf NF2 überprüft werden, da 2 oder mehr Attribute den Primärschlüssel bilden ✓
- ☒ first_name gehört nur zu einem Teil des PK (lect_ID), daher wird eine neue Tabelle erstellt, die lect_ID als PK und first_name als zusätzliches Attribut hat. first_name wird dann aus der Originaltabelle gelöscht ✓
- ☒ first_name, surname, sub_description und is_speaker sind Nicht-PK-Attribute und müssen jeweils auf ihre funktionalen Abhängigkeiten zum gesamten PK oder nur einem Teil davon überprüft werden ✓
- ☐ is_speaker ist funktional abhängig vom gesamten PK. Daher bleibt es in der Originaltabelle, wenn auf NF2 geprüft wird
- ☒ NF2 ergibt 3 Tabellen: lecturer (lect_ID, first_name, surname); subject(subject, sub_description, is_speaker); college(lect_ID, subject) ✓
- ☐ NF2 ergibt 2 Tabellen: lecturer (lect_ID, first_name, surname); subject(subject, sub_description, is_speaker);
- ☒ Die Namen der abgespaltenen Tabellen sind frei wählbar ✓
- ☒ NF2 ist hier auch in NF3 ✓
- ☒ Wenn eine Tabelle aus keinem oder nur einem Nicht-PK-Attribut besteht und NF2 hat, ist sie automatisch in NF3 ✓
- ☐ is_speaker hängt von lect_ID ab und ist somit transitiv abhängig vom Fach, muss also in NF3 abgespalten werden
- ☒ Für NF3 müssen alle Nicht-Schlüsselattribute auf Abhängigkeiten untereinander geprüft werden ✓
- ☒ Wenn ein Nichtschlüsselattribut von einem anderen Nichtschlüsselattribut abhängt (transitive Abhängigkeit), dann wird eine neue Tabelle mit dem zweiten Attribut als PK und dem ersten Attribut als zusätzliches Attribut erstellt. Das erste Attribut wird dann aus der Originaltabelle gelöscht, das zweite Attribut bleibt als Fremdschlüssel für die neue Tabelle. ✓
- ☐ Wenn in NF3 eine transitive Abhängigkeit erkannt wird, kann es nie passieren, dass eine neue Tabelle erstellt werden muss.
- ☐ NF3: first_name hängt vom surname ab, der von lect_ID abhängt

Frage 4

Teilweise richtig

Erreichte Punkte 3,43 von 4,00

v1 (neueste)

Geben Sie die richtigen Antworten zur Normalisierung ein:

Wählen Sie eine oder mehrere Antworten:

- ☒ Beim Normalisierungsprozess werden Tabellen aufgeteilt, um Probleme zu vermeiden ✓
- ☐ Wenn Sie Tabellen in NF3 haben, müssen Sie diese erneut auf NF2 und NF1 überprüfen.
- ☒ Das Zusammenführen bereits getrennter Tabellen sollte einfach sein und muss zu voll funktionsfähigen Tabellen mit derselben semantischen Bedeutung führen. ✓
- ☒ Bei der Normalisierung werden Attribute auf ihre Abhängigkeiten untereinander geprüft. ✓
- ☒ Tabellen in NF1 sind ein absolutes Muss, damit ein relationales Datenbankmodell überhaupt funktioniert. ✓
- ☒ In der Praxis müssen Tabellen nur für NF1, NF2 und NF3 überprüft werden. ✓
- ☒ Wenn Sie Tabellen in NF3 haben, können Sie sicher sein, dass in den Tabellen absolut keine Redundanz mehr vorhanden ist. ✗
- ☒ NF5 ist die höchste verfügbare NF. ✓
- ☒ Bei der Normalisierung muss man sich der semantischen Bedeutung von Attributen bewusst sein. ✓
- ☐ Bei der Normalisierung werden manchmal zwei oder mehr Tabellen zu einer einzigen Tabelle zusammengeführt

Frage 5

Richtig

Erreichte Punkte 5,00 von 5,00

v1 (neueste)

Richtige Antworten markieren:

Wählen Sie eine oder mehrere Antworten:

- ☐ Normalisierung hilft gegen fehleranfällige Tabellen, erhöht aber die Anzahl der Anomalien
- ☐ Um Anomalien zu vermeiden, sollten mehrere Tabellen zu einer einzigen Tabelle zusammengeführt werden
- ☒ Eindeutigkeitsbedingung (Entity-Integrity): Jede Tabelle hat einen Identifikationsschlüssel (Primärschlüssel), der jeden Datensatz eindeutig identifiziert ✓
- ☒ Referentielle Integrität: Jeder Wert eines Fremdschlüssels muss effektiv als Primärschlüsselwert in der referenzierten Tabelle vorhanden sein ✓
- ☒ Einfüge-Anomalie: Beim Einfügen neuer Datensätze werden fehlende Attribute mit NULL-Werten gefüllt (was bei Primärschlüsseln ein Problem darstellt) ✓
- ☒ Lösch-Anomalie: Das Löschen eines Wertes führt zum unerwünschten Verlust von Daten. ✓
- ☐ Kombinieren Sie Attribute aus mehreren Entitätstypen zu einer einzigen Beziehung, um einfacher Abfragen zu ermöglichen.
- ☒ Update-Anomalie: Die Änderung eines Attributwerts führt zu inkonsistenten Daten. ✓
- ☒ Wertbereichsbedingung (Semantische Integrität): verhindert falsch geschriebene Attributwerte ✓
- ☒ Ein Attribut einer Tabelle wird als redundant bezeichnet, wenn es ohne Informationsverlust weggelassen werden kann. ✓
- ☐ Referenzielle Integrität: Vermeiden Sie nach Möglichkeit, dass Fremdschlüssel auf Primärschlüssel anderer Tabellen verweisen
- ☒ Entwerfen Sie ein Beziehungsschema so, dass seine Bedeutung leicht erklärt werden kann (semantische Bedeutung sollte klar sein). ✓
- ☒ Vermeiden Sie nach Möglichkeit, Attribute in einer Basisrelation zu platzieren, deren Werte häufig NULL sein können. ✓
- ☒ Entwerfen Sie die Basisschemata so, dass in den Beziehungen keine Anomalien beim Einfügen, Löschen oder Ändern vorhanden sind. ✓