

Lösungsvorschlag:

1.

Objekttypen:

Kandidat (Attribute Id (P.K.), Name, Adresse, Telefonverbindung)

Test Zentrum (Attribute Id (P.K.), Name, Adresse, Telefonverbindung, Anzahl Testplätze)

Test Typ (Attribute Id (P.K.), Name (A.K.), Zeitdauer)

Notenliste kann als existenzabhängig von Test Typ gesehen werden

=>

Existenzabhängiger Objekttyp Notenliste (Attribute Test Typ Id (F.K. -> Test Typ), Note (zusammen P.K.), minimale Punktzahl)

Modellierung als Wiederholungsgruppe ist gefährlich, wenn sich das Benotungssystem ändert!

(ternärer) Beziehungstyp:

Kandidat – Test Typ – Test Zentrum (Kandidat legt Test eines Test Typs in einem Test Zentrum ab)

=>

assoziativer Objekttyp Test (Kandidat Id (F.K. -> Kandidat), Test Typ Id (F.K. -> Test Typ), Test Zentrum Id (F.K. -> Test Zentrum) Datum, Zeitpunkt des Beginns, Test Platz, Note)

Schlüsselkandidat:

Es ist davon auszugehen, daß ein Kandidat zu einem Zeitpunkt nur einen Test ablegen kann.

Daher bietet sich als Schlüsselkandidat an:

- Kandidat Id, Datum, Zeitpunkt des Beginns

2. Relationale Algebra

- a. Restriktion, Tabelle Professoren; Bedingung: Raum=1
- b. Restriktion, Tabelle Assistenten; Bedingung: Fachgebiet='Ideenlehre'
Natural Join mit Tabelle Professoren (Assistenten.Boss=Professoren.PersNr)
Restriktion; Bedingung Professoren.Name='Sokrates'
Projektion auf Assistenten.PersNr (oder Assistenten.*)
- c. Restriktion, Tabelle Vorlesungen; Bedingung: Name='Wissenschaftstheorie'
Natural Join mit Tabelle Professoren (Vorlesungen.gelesenVon=Professoren.Persnr)
Projektion auf Professoren.Name
- d. Restriktion, Tabelle Vorlesungen; Bedingung: Name='Wissenschaftstheorie'
Natural Join mit hoeren (Vorlesungen.VorlNr = hoeren.VorlNr)
Natural Join mit pruefen (hoeren.VorlNr = pruefen.VorlNr und hoeren.MatrNr = pruefen.MatrNr)
Projektion hoeren.MatrNr (bzw. pruefen.MatrNr)
- e. Resultat= Resultat1 minus Resultat2
Resultat1:
Restriktion, Tabelle Vorlesungen; Bedingung: Name='Wissenschaftstheorie'
Natural Join mit hoeren (Vorlesungen.VorlNr = hoeren.VorlNr)
Projektion hoeren.MatrNr
Resultat2:
Restriktion, Tabelle Vorlesungen; Bedingung: Name='Wissenschaftstheorie'
Natural Join mit pruefen (Vorlesungen.VorlNr = pruefen.VorlNr)
Projektion pruefen.MatrNr
- f. Es genügt zu überprüfen, ob die Vorlesung Wissenschaftstheorie Vorgänger der Stufe 2 hat:
Restriktion, Tabelle Vorlesungen; Bedingung: Name='Wissenschaftstheorie'
Natural Join mit voraussetzen, Rolle v1 (Vorlesungen.VorlNr = v1.Nachfolger)
Natural Join mit voraussetzen, Rolle v2 (v1.Vorgaenger = v2.Nachfolger)

SQL:

- a.

```
select *
from Professoren
where Raum=1;
```
- b.

```
select t1.*
from Assistenten t1, Professoren t2
where t1.Fachgebiet='Ideenlehre'
and t2.Name='Sokrates'
and t1.Boss=t2.PersNr;
```
- c.

```
select distinct t2.name
from Vorlesungen t1, Professoren t2
where t1.gelesenVon=t2.Persnr
and t1.Name='Wissenschaftstheorie';
```
- d.

```
select distinct t2.MatrNr
from Vorlesungen t1, hoeren t2, pruefen t3
where t1.VorlNr=t2.VorlNr
and t2.VorlNr=t3.VorlNr
and t2.MatrNr=t3.MatrNr
and t1.Name='Wissenschaftstheorie';
```
- e.

```
select distinct t2.MatrNr
from Vorlesungen t1, hoeren t2
where t1.VorlNr=t2.VorlNr
and t1.Name='Wissenschaftstheorie'
where t2.MatrNr not in
(
select distinct t4.MatrNr
from Vorlesungen t3, pruefen t4
where t3.VorlNr=t4.VorlNr
and t3.Name='Wissenschaftstheorie'
)
oder entsprechendes „minus“ Konstrukt
```
- f.

```
select count(*)
from Vorlesungen t1, voraussetzen t2, voraussetzen t3
where t1.Name='Wissenschaftstheorie'
and t1.VorlNr=t2.Nachfolger
and t2.Vorgaenger=t3.Nachfolger;
```

Ergebnis = 0 => Man kann die Vorlesung im 2.Semester hören

Ergebnis > 0 => Man kann die Vorlesung erst im 3.Semester hören

3.

- Wenn man davon ausgeht, daß Adressen als atomar betrachtet werden können, ist die 1NF als Voraussetzung für 2NF und damit 3NF erfüllt, ansonsten kann man die Betrachtung hier abbrechen.
- Da der Primärschlüssel aus 1 Attribut besteht, ist ein Verstoß gegen die 2NF ausgeschlossen.
- Wenn man davon ausgeht, daß die Geschäftsadresse nicht von der Abteilung abhängig ist, ist 3NF erfüllt.

Falls das Unternehmen nur wenige Standorte hat, kann man sich trotzdem überlegen, die Geschäftsadressen = Adressen der Standorte in eine eigene Tabelle auszulagern. Wenn man dann den Primärschlüssel dieser Tabelle, die Geschäftsadressen Id, in die Tabelle mitarbeiter aufnimmt, so liegt danach ein Verstoß gegen 3NF vor, den man auflöst, indem man die Adreßinformationen in der eigenen Adreßtabelle verwaltet und in der mitarbeiter Tabelle nur die Geschäftsadressen Id führt.

4.

Die Grundlegenden Integritätsbedingungen sind:

Entity Integrity:

- Kein Attribut, das Bestandteil eines Schlüsselkandidaten ist, darf den Wert NULL annehmen

Referential Integrity:

- Ein Fremdschlüsselwert kommt im zugehörigen Schlüsselkandidaten vor, oder er ist komplett NULL

(Diese Regeln besagen insbesondere, daß es möglich ist, die Konzepte Schlüsselkandidat und Fremdschlüssel abzubilden, was über primary key, unique und foreign key constraints möglich ist.)

Business Rules:

Das Datenbanksystem ermöglicht es, Bedingungen an die Daten, die aus dem Geschäftsprozeß heraus entstehen, abzubilden. Dazu können check constraints (deklarativ, wirksam zum Zeitpunkt der Definition) und Trigger (wirksam erst ab dem Zeitpunkt der Definition) eingesetzt werden.

5. Das Konzept der Datenunabhängigkeit basiert auf verschiedenen Sichten in einer 3 Schichten Architektur, die man auf die Daten haben kann:

- die (untere) physische Sicht, die sich mit den Themen physische Ablage der Daten, Datenzugriff, Performance ... beschäftigt
- die (mittlere) logische Sicht, die sich mit der gesamtheitlichen Betrachtung der Daten (und Datenzugriffe) beschäftigt
- die (obere) Anwender- bzw. Entwicklersicht, die dem Anwender des Datenmodells (i.a. dem Anwendungsentwickler) den Ausschnitt aus den Daten bzw. die Zugriffsmöglichkeiten zur Verfügung stellt, die er benötigt.

Unter Datenunabhängigkeit versteht man die Unabhängigkeit der Anwendersicht von der physischen Sicht (physische Datenunabhängigkeit) und der logischen Sicht (logische Datenunabhängigkeit). Logische Datenunabhängigkeit soll dadurch erreicht werden, daß die logische Sicht durch Konstrukte wie Views, Synonyme, Stored Procedures ... auf die Anwendersicht abgebildet wird. Physische Datenunabhängigkeit soll dadurch erreicht werden, daß die Anwendersicht durch die logische Sicht von der physischen Schicht abgeschirmt ist.

Die Formulierung „soll“ besagt, daß die Anforderungen im Hinblick auf die Datenunabhängigkeit nicht 100 %ig umgesetzt sind.