Functionele afhankelijkheden & normalisatie I

Informele richtlijnen voor een goed ontwerp

1. Ontwerp een relatieschema zo dat zijn betekenis gemakkelijk verklaard kan worden
2. Ontwerp een relatieschema zo dat redundantie vermeden wordt en geen toevoeg-, weglaat- of wijziging-anomalieën kunnen voorkomen.
    a. **Insertion Anomalies** happen when inserting vital data into the database is not possible because other data is not already there. For example, if a system is designed to require that a customer be on file before a sale can be made to that customer, but you cannot add a customer until they have bought something, then you have an insert anomaly. It is the classic "catch-22" situation.
    b. **Deletion Anomalies** happen when the deletion of unwanted information causes desired information to be deleted as well. For example, if a single database record contains information about a particular product along with information about a salesperson for the company and the salesperson quits, then information about the product is deleted along with salesperson information.
    c. **Update Anomalies** happen when the person charged with the task of keeping all the records current and accurate, is asked, for example, to change an employee's title due to a promotion. If the data is stored redundantly in the same table, and the person misses any of them, then there will be multiple titles associated with the employee. The end user has no way of knowing which is the correct title.
3. Vermijd zoveel mogelijk attributen waarvan de waarden nul kunnen zijn.
4. Ontwerp relatieschema's zo dat ze na een equi-join op gerelateerde attributen geen onechte tupels opleveren.

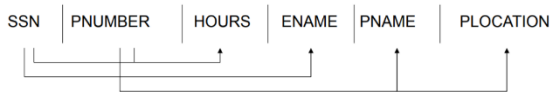
Functionele afhankelijkheden

Informele definitie: If column A of a table uniquely identifies the column B of same table then it can represented as A $\rightarrow$ B (Attribute B is functionally dependent on attribute A)

Formele definitie: Zij gegeven een relatieschema R = {A_1, A_2, ..., A_n} en attributenverzamelingen X en Y. We zeggen dat Y functioneel afhankelijk is van X, genoteerd X $\rightarrow$ Y, als en slechts als

1. $X \neq \emptyset$
2. $\forall$ mogelijke extensies r van R: $t_1, t_2 \in r$ en $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$

Voorbeeld



- Relatie EMP_PROJ
    - SSN $\rightarrow$ ENAME
    - PNUMBER $\rightarrow$ { PNAME, PLOCATION }
    - { SSN,PNUMBER } $\rightarrow$ HOURS

Puntje 2 van formele definitie: Als verschillende SSN en PNAME gelijk zijn aan elkaar dan moet hun HOURS gelijk zijn.

Afleidingsregels zijn nodig voor alle functionele afhankelijkheden af te kunnen leiden uit een verzameling.

- FD1. reflexiviteitsregel: $Y \subseteq X \Rightarrow X \rightarrow Y$
- FD2. uitbreidingsregel: $\{ X \rightarrow Y \} \models XZ \rightarrow YZ$
- FD3. transitiviteitsregel: $\{ X \rightarrow Y, Y \rightarrow Z \} \models X \rightarrow Z$
- FD4: decompositieregel $\{ X \rightarrow YZ \} \models X \rightarrow Y$
- FD5: verenigingsregel $\{ X \rightarrow Y, X \rightarrow Z \} \models \{ X \rightarrow YZ \}$
- FD6: pseudo-transitiviteitsregel $\{ X \rightarrow Y, WY \rightarrow Z \} \models WX \rightarrow Z$

Voorbeeld

F = {Ssn → {Ename, Bdate, Address, Dnumber}, Dnumber → {Dname, Dmgr_ssn}}

Functionele afhankelijkheden afgeleid uit F

1. Ssn → {Dname, Dmgr_ssn}
2. Ssn → Ssn
3. Dnumber → Dname

XF^+ ("sluiting van X t.o.v. F"): verzameling van alle attributen die functioneel bepaald zijn door X. Hiervoor kan je de algoritme gebruiken op dia 31.

Voorbeeld

- F = {SSN → ENAME, PNUMBER → { PNAME, PLOCATION }, { SSN, PNUMBER } → HOURS }
- Sluitingen t.o.v. F zijn dan:
  - $\{ SSN \}^+ = \{ SSN, ENAME \}$
  - $\{ PNUMBER \}^+ = \{ PNUMBER, PNAME, PLOCATION \}$
  - $\{ SSN, PNUMBER \}^+ = \{ SSN, PNUMBER, ENAME, PNAME, PLOCATION, HOURS \}$

Overdekking & Equivalentie

- Zij E en F verzamelingen van functionele afhankelijkheden van R
- E wordt overdekt door F a.s.a. $E \subseteq F^+$
  - alle functionele afhankelijkheden in E volgen uit F via afleidingsregels
  - automatisch ook $E^+ \subseteq F^+$
- E en F zijn equivalent a.s.a. $E^+ = F^+$
  - E overdekt F en F overdekt E
  - $E^+ \subseteq F^+$ en $F^+ \subseteq E^+ \Rightarrow E^+ = F^+$
- Nagaan of F E overdekt:
  - voor alle $X \rightarrow Y$ in E: bepaal X_F^+ en controleer of $Y \subseteq X_F^+$

Voorbeeld

Overdekt F de verzameling E?

- E = {A → BC, D → AE}
- F = {A → B, AB → C, D → AC, D → E}
  - $A \rightarrow BC$ & $A_F^+ = \{A, B, C\}$ & $BC \subseteq \{A, B, C\}$
  - $D \rightarrow AE$ & $D_F^+ = \{D, A, C, E, B\}$ & $AE \subseteq \{D, A, C, E, B\}$

- Overdekt E de verzameling F?
  - $A \rightarrow B$ & $A_E^+ = \{A, B, C\}$ & $B \subseteq \{A, B, C\}$
  - $AB \rightarrow C$ & $AB_E^+ = \{A, B, C\}$ & $C \subseteq \{A, B, C\}$
  - $D \rightarrow E$ & $D_E^+ = \{D, E, A, C, B\}$ & $E \subseteq \{D, E, A, C, B\}$
  - $D \rightarrow AC$ & $D_E^+ = \{D, A, E, B, C\}$ & $AC \subseteq \{D, A, E, B, C\}$
- Conlusie: E overdekt F en F overdekt E, dus E en F zijn equivalent.

Minimale overdekking

Informele algoritme

1. Indien rechterlid niet uit 1 attribuut bestaat, moet je opsplitsen.
   Bv. $D \rightarrow AC$ wordt $D \rightarrow A$ & $D \rightarrow C$.
2. Als linkerkant meer dan 1 attribuut heeft, kunnen we er dan 1 weglaten?
   Bv. $AB \rightarrow C$ moet $B \rightarrow C$ worden. Daarna kijken of $B \in A_F^+$. Indien dit klopt, kan je B weghalen uit $AB \rightarrow C$.
3. Kijken of je 1ste, 2de of een van de andere regels kan weghalen.
   Bv. $B \rightarrow A$ weglaten uit F. Kijk of $A \in B_F^+$.

Normaalvormen

Oplossing voor

1. Redundantie van gegevens in een extensie
2. Problemen bij het onderhoud

Hoe hoger de normaalvorm, des te minder afhankelijkheden zich kunnen voordoen. (hoe hoger, hoe beter dus)

Enkele definities voordat we normaalvormen bespreken

- **Definitie**: K is een **supersleutel** voor een relatie R met schema SR = <U,F> a.s.a.
  $K \subseteq U$ en $K \rightarrow U \in F^+$.
- **Definitie**: K is een **sleutel** of **kandidaatsleutel** voor een relatie R met schema SR = <U,F> a.s.a.
  K een supersleutel is voor R en geen enkele echte deelverzameling van K een supersleutel is voor R. (Candidate key is a super key from which you cannot remove any fields. If you remove it, the combination will not be unique)
- **Definitie**: een attribuut A is een **sleutelattribuut** voor een relatie R met schema SR = <U,F> a.s.a.
  er bestaat een sleutel K voor R waarvoor geldt $A \in K$.

0de normaalvorm

→ Mogen samengesteld of meerwaardig zijn.

1ste normaalvorm

→ Each column of your table should be single valued which means they should not contain multiple values.

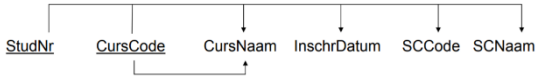
Oplossing

1. Samengesteld attribuut in meerdere attributen opsplitsen.
2. Voor meerwaardig attribuut meerdere tupels voorzien of nieuwe relatie creëren met zelfde sleutel.

2de normaalvorm

→ The table should be in the First Normal Form and there should be no partial dependency.

Aanloop definitie: Als X → Y, dan zeggen we dat Y partieel functioneel afhankelijk is van X indien er een Z ⊂ X bestaat zodat Z → Y; anders noemen we Y volledig functioneel afhankelijk van X.



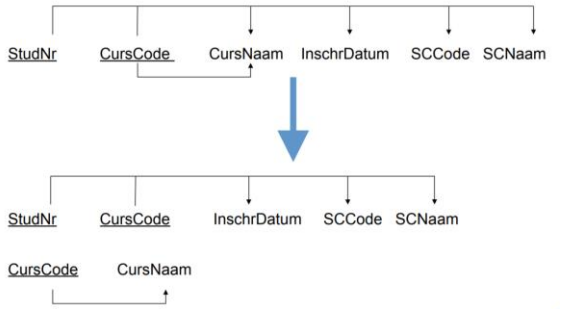
Voorbeeld

Informal definition: Partial dependency happens when an attribute in a table depends on only a part of the primary key and not on the whole key.

Oplossing

To remove partial dependency, we can divide the table, remove the attribute which is causing partial dependency, and move it to some other table where it fits in well.

Voorbeeld

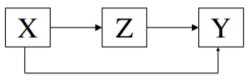


3de normaalvorm

→ The table should be in the Second Normal form and it should not have transitive dependency.

Aanloop definitie: Een functionele afhankelijkheid X → Y is een transitieve functionele afhankelijkheid a.s.a. er een Z bestaat zodat volgende 3 voorwaarden voldaan zijn:

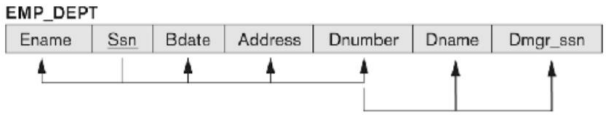
1. Z is volledig en niet-triviaal functioneel afhankelijk van X.
2. Z is geen (echte of onechte) deelverzameling van een kandidaatsleutel.
3. Y is niet-triviaal functioneel afhankelijk van Z. We zeggen dat Y transitief functioneel afhankelijk is van X via Z.



X bevat dus de volledige key, Z mag geen deelverzameling zijn van een kandidaatsleutel en Y mag ook geen deelverzameling zijn van een kandidaatsleutel.

Informal definition: Transitive dependency takes place when a non-prime attribute depends on other non-prime attributes rather than depending upon the prime attributes or primary key.

Voorbeeld X = Ssn, Z = Dname, = Dmgr_ssn



Oplossing

Remove the non-prime keys that are transitive dependent and put them in a new table.

Voorbeeld