

Hoofdstuk 5. Logisch gegevensmodel

5.1 Probleemstelling

In de vorige hoofdstukken kwamen diverse gegevensanalyses aan bod. Met behulp van DFD worden de datastromen tussen data stores, processen en externe entiteiten voorgesteld. Om problemen bij de opslag en het onderhoud van de gegevens zo optimaal mogelijk te laten verlopen wordt de normalisering van de gegevens uitgevoerd. Vervolgens wordt de samenhang tussen de data stores weergegeven via ERD.

De effectieve implementatie van de gegevens is nog niet mogelijk, omdat nog een aantal schakels ontbreken:

- er is niet expliciet aangegeven hoe entiteiten met foreign keys gekoppeld zijn met een primary key;
- het datatype, zoals bijvoorbeeld integer of tekst, is niet aangeduid.

Om aan deze tekorten tegemoet te komen wordt een logisch gegevensmodel opgesteld. Dit vormt de laatste stap voor de effectieve implementatie.

5.2 Mogelijke voorstelling

Er zijn verschillende manieren om het logisch gegevensmodel voor te stellen. Zoals reeds in Hoofdstuk 4 vermeld, kunnen ook attributen getoond worden in een ERD en kunnen eveneens in een ERD de elementen van het logisch gegevensmodel voorgesteld worden. Dat is een subjectieve keuze.

5.2.1 Tabel met attributen

In het volgende voorbeeld worden twee tabellen met attributen voorgesteld. Boven de tabel wordt in het vet de naam van de tabel geplaatst met daaronder de attributen. De naam van de tabel begint altijd met een hoofdletter en de attributen starten met een kleine letter, maar kunnen hoofdletters bevatten. De naam van een attribuut mag geen spaties bevatten. Indien een attribuut door meerdere woorden wordt omschreven, wordt gebruik gemaakt van een underscore ofwel een CamelCase. Een CamelCase betekent dat samengestelde woorden of zinnen, die normaliter met een spatie gescheiden worden, geschreven worden als één woord door de spatie en de daaropvolgende letter te vervangen door de corresponderende hoofdletter. Een consequente toepassing van ofwel de underscore ofwel de CamelCase is vereist!

Klanten	Postcodes
familienaam	postcode
voornaam	gemeente
straat	
huisnummer	
postcode	
klantId	

5.2.2 Sleutelwaarden

In de tabel wordt de primary key bovenaan in de tabel geplaatst en onderlijnd. Achter een foreign key wordt FK en een pijl geplaatst, gevolgd door een verwijzing naar de primary key. Deze verwijzing bevat de tabelnaam en tussen ronde haken de overeenstemmende primary key.

Klanten	Postcodes
<u>klantId</u>	<u>postcode</u>
familienaam	gemeente
voornaam	
huisnummer	
postcode	FK⇒Postcodes(postcode)

5.2.3 Metadata

Metadata zijn beschrijvingen van de data en geven in dit geval informatie over de attributen. Achter een attribuut wordt daartoe een dubbele punt geschreven, gevolgd door de metadata.

5.2.3.1 Standaard types

Hier volgen een aantal standaard types voor metadata die ook gebruikelijk zijn bij programmeertalen.

Integer	Natuurlijk getal
Decimal	Decimaal getal
String	Tekst
Date	Datum (zonder tijd)
Time	Tijd (zonder datum)
Datetime	Datum+ tijd

Er bestaan meerdere wijzen om standaard types aan te duiden. Het is van belang om de gekozen wijze consequent te gebruiken.

Het logisch gegevensmodel mag niet specifiek gelinkt worden aan een implementatie, omdat het model onafhankelijk dient te zijn van de implementatie.

Voor de datatypes gelden een aantal afspraken. Datatypes beginnen met een hoofdletter en de maximale lengte kan aangeduid worden. In het voorbeeld bevat een klantId maximaal 8 karakters en de postcode (in België) bestaat uit (maximaal) 4 cijfers.

Het type Decimal wordt weergegeven als Decimal(a,b), waarbij a het aantal cijfers en b het aantal na de komma aanduidt.

In het datatype kan een minimum en een maximum aantal tekens worden aangegeven. In het voorbeeld kan de voornaam uit minstens 2 en ten hoogste 30 karakters bestaan.

Klanten <u>klantId</u> : String(8) familienaam: String(30) voornaam: String(2..30) huisnummer: Integer(4) postcode: Integer(4)	Postcodes <u>postcode</u> : Integer(4) gemeente: String(35)
--	--

FK⇒Postcodes(postcode)

Het type van een foreign key moet altijd gelijk zijn aan de verwezen primary key! In het voorbeeld komt dit tot uiting bij het datatype van postcode, dat zowel in de foreign key als primary key als een Integer met maximum 4 cijfers aangeduid wordt.

5.2.3.2 Set Data

Als een attribuut slechts een bepaald aantal waarden kan aannemen, worden tussen accolades alle mogelijk waarden opgesomd of wordt een range van waarden aangeduid.

Patiënt <u>patientId</u> : String(8) familienaam: String(30) voornaam: String(2..30) geboorteDatum: Date lengte: Integer(3), {0..300} gewicht: Decimal(5,3) geslacht: String(5), {man,vrouw} oogkleur: String(6), {blauw,groen,bruin,albino}

5.2.3.3 Null / Not Null

Hiermee kan je aangeven of een veld leeg mag gelaten worden of niet. We duiden enkel Not Null aan voor een veld. Wanneer er niets wordt aangeduid, wordt verondersteld dat het veld null waarden kan bevatten. Een PK is automatisch Not Null. Dit moeten we niet expliciet meer aangeven.

PatiëntpatientId: String(8)

familienaam: String(30), not null

voornaam: String(30), not null

geboorteDatum: Date, not null

lengte: Integer(3), {0..300}

gewicht: Decimal(5,3)

geslacht: String(5), {man,vrouw}, not null

oogkleur: String(6), {blauw,groen,bruin,albino}

Bibliografie

- [1] Charles William Bachman. „Data structure diagrams”. In: DataBase (1969), p. 4-10.
- [2] Guido Bakema, Jan Pieter Zwart en Harm van der Lek. Fully Communication Oriented Information Modeling (FCO-IM). Bommelj’e Cromptvoets en partners (BCP), 2002.
- [3] PeterPin shan Chen. „The Entity-RelationshipModel: Towarda UnifiedView of Data”. In: ACM Transactions on Database Systems 1 (1976), p. 9-36.
- [4] Edgar Frank Codd. „A RelationalModel of Data for Large Shared Data Banks”. In: Commun. ACM 13.6 (1970), p. 377-387.
- [5] Isabelle Godfrind. Dataorganisatie fundamentals. Xios Hogeschool Limburg, 2012.
- [6] Instruct. Informatica Online. Dec 2007. url : <http://www.instruct-online.nl/>.
- [7] James Martin en Clive Finkelstein. Information Engineering. Savant institute, 1981.
- [8] Morgan Masters. An Introductionto Data Flow Diagrams. Jan 2012. url : <http://www.modernanalyst.com/Resources/Articles/tabid/115/ArticleType/ArticleView/ArticleId/2009/An-Introduction-to-Data-Flow-Diagrams.aspx>.
- [9] Ronald G. Ross. Entity modeling: techniques and application. Database Research Group, 1987. isbn: 9780941049009. url : <http://books.google.be/books?id=JqoZAQAIAAJ>.
- [10] James Rumbaugh. „A ParallelAsynchronous Computer Architecturefor Data Flow Programs”. ProefschriftM.I.T. Laboratoryfor ComputerScience, Cambridge,MA, mei 1975.
- [11] Ed Yourdon. Structured Analysis: Dataflow Diagrams. Jun 2011. url : http://www.yourdon.com/structuredanalysis/wiki/index.php?title=Chapter_9.

