

Relationele databases

SqlServer en Oracle zijn relationele client server databases. De verwerking van de opdrachten vindt plaats op de server. Access is een *relationele* pc database. De verwerking van de gegevens vindt op de pc plaats.

Een *relationele* database is gebaseerd op de leer der verzamelingen. De gegevens worden weergegeven in *tabellen*. Om dubbele gegevens te vermijden worden gegevens in verschillende tabellen opgeborgen. Deze tabellen staan in een bepaalde relatie tot elkaar. Een tabel bevat gelijksoortige gegevens die zijn verdeeld over *kolommen (velden)*. Elk *record* in een tabel moet uniek zijn. De *primaire sleutel* bevat dan ook unieke waarden. Een primaire sleutel bestaat uit een of meer velden, b.v. klantnr, of postcode en huisnummer. Wanneer een relatie tot een andere tabel bestaat dan komt de *primaire sleutel* in de ene tabel overeen met een *vremde sleutel* in de andere tabel.

Gegevens worden uit een relationele database opgevraagd met de vraagtaal *SQL*. Deze taal kan met eenvoudige opdrachten gegevens uit een of meer tabellen opvragen, statistische gegevens berekenen en gegevens groeperen en sorteren. Ook kan met *SQL* gegevens worden gewijzigd, toegevoegd en verwijderd, tabellen worden aangemaakt en verwijderd, beheerfuncties worden uitgevoerd. *SQL* bestaat in verschillende versies. Elke database leverancier heeft een *SQL* versie volgens de ANSI standaard met meestal toegevoegd extra functionaliteit. De versies verschillen dan ook in lichte mate.

Entiteiten

Bij de keuze voor de gegevens die een organisatie wil bewaren wordt gebruik gemaakt van de begrippen *entiteit*, *entiteitstype*, *attribuut* en *attribuuttype* om de relatie tussen de gegevens en de werkelijkheid vast te leggen.

Entiteit

Zaken en gebeurtenissen waarvan gegevens moeten worden vastgelegd worden een entiteit genoemd: b.v. mens, voorstelling, auto.

Entiteitstype

Een specifiek type van entiteit. Zo kan van de entiteit mens typen worden onderscheiden als patient, docent, cursist, bestuurder, medewerker waarbij verschillende gegevens van belang zijn:
patient (naam, adres, pc, plaats, leeftijd, gewicht, erfelijke afw, medicijngebruik, huisarts)
docent (naam, adres, pc, plaats, kennis, aantal dagen, salaris, vervoer)

Attribuut

Een attribuut is een eigenschap van een entiteit. Van de mens is een arm een attribuut, maar ook de naam en een karaktereigenschap.

Attribuuttype

De specifieke eigenschap die moet worden vastgelegd is het attribuuttype. Van een patient b.v. de bloeddruk.

Systemanalyse

Om te komen tot de juiste gegevens in een informatiesysteem wordt een aantal stappen uitgevoerd:

Objectanalyse (wat is belangrijk voor een organisatie: producten, rapporten, overzichten)

Functieanalyse (inzicht in de structuur van een organisatie, uitwisseling van gegevens gekoppeld met activiteiten en entiteiten, eventueel gebruikmakend van bestaande informatieuitwisseling als documenten, formulieren, rapporten)

Entiteitanalyse (klant, mogelijke klant, klacht, en per entiteit de noodzakelijke gegevens)

Relationeel Database Ontwerp

Gegevensanalyse (structuur aanbrengen in de gegevens, groeperen met een zo groot mogelijke samenhang, hier kan de methode van het *Normaliseren* voor worden gebruikt)

Een gegevensgroep/object kent meerdere occurrences of voorkomens.

Functionele relaties:

- *Enkelvoudige functionele afhankelijkheid*
Bij een voorkomen van een attribuut is slechts één voorkomen van een ander attribuut: $A=B$, (*ordernr* -> *orderdatum*)
- *Volledige functionele afhankelijkheid*
Bij een voorkomen van een combinatie van twee sleutelattributen is slechts één voorkomen van een ander attribuut: $C=A \text{ and } B$, (*ordernr*, *artnr*) -> *totaalprijs*)
- *Transitieve functionele afhankelijkheid*
Bij een voorkomen van een attribuut dat niet rechtstreeks functioneel afhankelijk is van de sleutel maar via een ander attribuut: $A=B=C$ (*ordernr* -> *levnr* -> *leveranciersnaam*)
- *Meervoudige afhankelijkheid*
Wanneer een sleutel meer dan drie attributen bevat, kan een sleutelattribuut functioneel afhankelijk zijn van de andere sleutelattributen maar niet van de unieke combinatie van de andere sleutelattributen: $C=A \text{ or } B$

Normaliseren

Opsplitsen van objecten zonder verlies aan informatie opdat het onderhoud en gebruik niet tot onregelmatigheden leidt.

E.F. Codd

5 normaalvormen

Voorbeeld:

Cursisten hebben een cursistnummer en een naam. Zij volgen een of meer cursussen. Een cursus heeft een cursuscode, een omschrijving. Voor het *object cursusinschrijving* is ook de cursusdatum en de plaats van belang.

De gegevens voor het object *cursusinschrijving*:

(Cursistnr, cursistnaam, [cursuscode, omschrijving, cursusdatum, cursusplaats, plaatscode])

0^e normaalvorm

Cursistinschrijving

cursistnr
cursistnaam
cursuscode
omschrijving
cursusdatum
inschrijfdatum
plaatscode
cursusplaats

primaire sleutel: cursistnr

1^e normaalvorm

resultaat na afsplitsing van de *repeterende groep*

Cursist

cursistnr
cursistnaam

Cursistinschrijving

cursuscode

Relationeel Database Ontwerp

cursistnr
 cursusdatum
omschrijving
plaatscode
cursusplaats

primaire sleutel bij cursist object: cursistnr

primaire sleutel bij cursusinschrijving object: cursuscode, cursistnr en cursusdatum

2^e normaalvorm

resultaat van het afsplitsen van *gedeeltelijke sleutelafhankelijkheid* uit de 1^e normaalvorm zonder verlies aan informatie. Alle niet-sleutelattributen zijn volledig functioneel afhankelijk van de sleutel. Onderzoek van afsplitsing is alleen nodig bij samengestelde sleutels.

Cursist

cursistnr
cursistnaam

Cursistinschrijving

cursuscode
 cursistnr
 cursusdatum
plaatscode
cursusplaats

Cursus

cursuscode
omschrijving

primaire sleutel bij cursist object: cursistnr

primaire sleutel bij cursusinschrijving object: cursuscode, cursistnr en cursusdatum

primaire sleutel bij cursus object: cursuscode

3^e normaalvorm

Resultaat na afsplitsing van *indirecte sleutelafhankelijkheid*. Het gevolg is dat elk attribuut direct afhankelijk is van de sleutel.

Cursist

cursistnr
cursistnaam

Cursistinschrijving

cursuscode
 cursistnr
 cursusdatum
 plaatscode

Cursus

cursuscode
omschrijving

Cursusplaats

plaatscode
cursusplaats

primaire sleutel bij cursist object: cursistnr

primaire sleutel bij cursusinschrijving object: cursuscode, cursistnr en cursusdatum

primaire sleutel bij cursus object: cursuscode

primaire sleutel bij cursusplaats object: plaatscode

4^e normaalvorm

Groeperen op basis van *meervoudige afhankelijkheid*. Sleutel met drie of meer velden. Cursistnr is afhankelijk van unieke combinatie, dus, 4^e is gelijk aan 3^e.

5^e normaalvorm

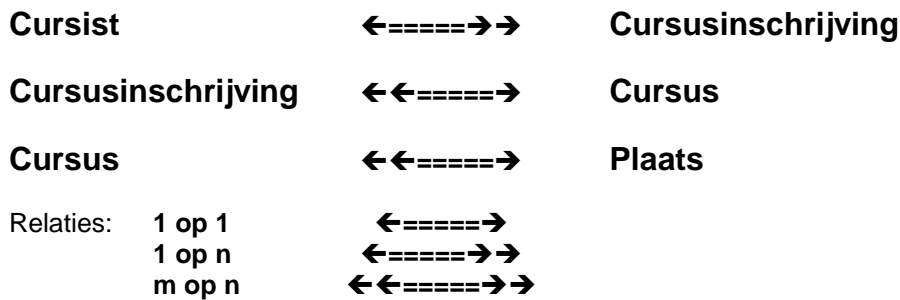
De genormaliseerde objecten worden weer geïntegreerd tot een logisch geheel.

Voordeel normalisatie:

- Geen informatieverlies
- Reductie van redundantie
- Minder ruimte nodig
- Drastische ingrepen als nieuwe objecten zijn gemakkelijker te realiseren

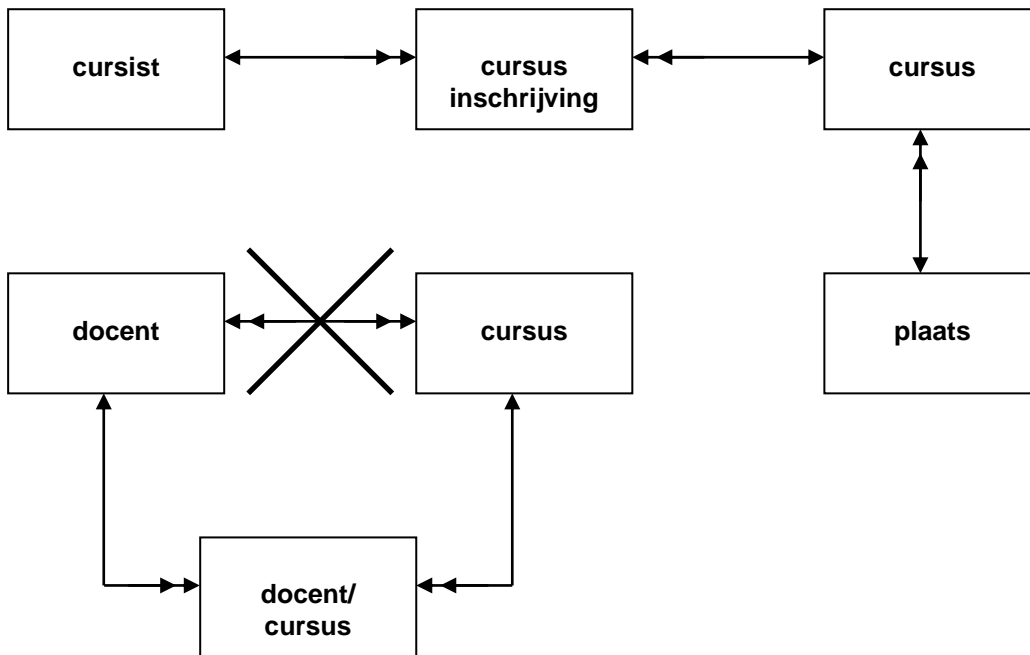
Gegevens Structuur Diagram

(Bachmann-diagram)



In Access altijd 1 op n relaties, geen m op n. Vervang een m op n relatie door twee 1 op n relaties.

tekening



Voorbeeld:

ordernummer : 8564

orderdatum: 12 jan 03

leveranciernummer: 532
naam: De Hamer
plaats: Amsterdam

leverdatum: 21 jan 03

te leveren artikelen

artikelnummer	artikelomschrijving	hoeveelheid	prijs
3217	bout A19	2300	€ 187,--
4179	moer M16	1000	€ 50,--
7235	spijker C12	3000	€ 45,--
			€ 282,--

1. Geef een beschrijving van de inkooporder (negeer de totaalprijs). Geef ook de sleutel en de repeterende groep aan.

2. Hoe is de afhankelijkheid tussen:

Ordernummer -> orderdatum
Artikelnummer -> hoeveelheid
(Ordernummer, artikelnummer) -> prijs
(Ordernummer, artikelnummer) -> artikelomschrijving
ordernummer -> leveranciersnaam

3. Hoe ziet de 1^e, 2^e en 3^e normaalvorm eruit?

Uitwerking vraag 2:

<i>Ordernummer -> orderdatum</i>	<i>(volledig functioneel afhankelijk)</i>
<i>Artikelnummer -> hoeveelheid</i>	<i>(niet functioneel afhankelijk. Verschillende waarden voor hoeveelheid bij hetzelfde artikelnr)</i>
<i>(Ordernummer, artikelnummer) -> prijs</i>	<i>(volledig functioneel afhankelijk)</i>
<i>(Ordernummer, artikelnummer) -> artikelomschrijving</i>	<i>(niet volledig functioneel afhankelijk, wel volledig functioneel afhankelijk van artikelnr)</i>
<i>ordernummer -> leveranciersnaam</i>	<i>(transitief functioneel afhankelijk, via leveranciernr)</i>