# 9. Meerdere tabellen

Na het bestuderen van dit hoofdstuk wordt van je verwacht dat je:

* weet wat een sleutel (primary key) is
* weet wat een vreemde sleutel (foreign key) is
* weet wat een relatie is

In de database is nog niet opgenomen aan welke opleidingen de studenten deelnemen. Om dat te kunnen doen moeten we eerst de opleidingen in de database opnemen. Dat doen we door een tweede tabel te maken met daarin de opleidingen. We gaan dus eerst een nieuwe tabel maken:

**create table opleiding (**

**opleidingscode varchar(3),**

**naam varchar(30),**

**niveau char(1),**

**primary key(opleidingscode)**

**)**

**engine = InnoDB;**

De meeste regels hierboven zijn duidelijk (Anders zie hoofdstuk 2). De regel met primary key is nog niet behandeld. Aan een sleutel kan de database het record in de tabel herkennen. Dat betekent dat het veld in de kolom van de tabel uniek moet zijn. Als er twee sleutels zijn met dezelfde waarde, kan de database de records (regels) niet meer herkennen. De database ziet er op toe dat de sleutels uniek zijn.

Opdracht 9.1

Voer de volgende query in:

insert into opleiding (opleidingscode, naam, niveau)

values (‘AO’, ‘ApplicatieOntwikkeling’, ‘4’);

Deze query gaat goed. Voer nu de volgende query in:

insert into opleiding (opleidingscode, naam, niveau)

values (‘AO’, ‘ApothekersOpleiding’, ‘4’);

We krijgen dan de volgende foutmelding:

ERROR 1062 (23000): Duplicate entry 'AO' for key 'PRIMARY'

Met andere woorden: De database accepteert geen tweede sleutel met dezelfde waarde. Wel accepteert de database de volgende query, omdat de sleutel een andere waarde heeft.

Opdracht 9.2

Voer de volgende query in:

insert into opleiding (opleidingscode, naam, niveau)

values (‘MBI’, ‘Medewerker Beheer ICT’, ‘3’);

Opdracht 9.3

Zorg ervoor dat de tabel opleiding de volgende inhoud heeft en zorg ervoor dat bekend is wat de sleutel is:

* AO, Applicatieontwikkeling, 4
* MV, MediaVormgeving, 4
* DT3, DeskTopPublising, 3
* DT2, DeskTopPublising, 2
* MBI, Medewerker beheer ICT, 3
* IB, ICT Beheerder, 4

Nu moeten we een manier vinden om in de database vast te leggen welke student welke opleiding of opleidingen volgt. Daarvoor kunnen we een veld opleiding toevoegen aan de tabel student. We zetten dan de sleutel van de opleiding in dat veld. Maar als de student van opleiding verandert, kunnen we dat niet vastleggen, want het veld is al bezet door de vorige opleiding. En omdat er maar een veld beschikbaar is, kan een student altijd maar een opleiding volgen. Wat ook niet kan is gegevens vastleggen die en met de student en met de opleiding te maken hebben, bijvoorbeeld wanneer is een student met een opleiding begonnen. Dat is dus niet handig.

Daarom hanteren we de volgende oplossing. We maken nog een tabel en die noemen we studentopleiding. In die tabel gebruiken we twee velden, het eerste veld is voor de sleutel van de student en het tweede veld is voor de sleutel van de opleiding. Deze sleutels noemen we vreemde sleutels (foreign keys) omdat deze sleutels afkomstig zijn uit andere tabellen, waar ze als eerste sleutels (primary keys) functioneren. Verder maken we nog twee velden aan met de namen startdatum en einddatum. Wanneer is de student begonnen met de opleiding en wanneer is hij gestopt met de opleiding. Tot slot maken we een veld aan met de naam diploma, waarin staat of de opleiding is afgesloten met een diploma of niet:

**create table studentopleiding (**

**studentnr varchar(5),**

**opleidingscode varchar(3),**

**startdatum date,**

**einddatum date,**

**diploma tinyint(1) default 0,**

**constraint foreign key(studentnr) references student(studentnr)**

**on delete no action**

**on update no action,**

**constraint foreign key(opleidingscode) references opleiding(opleidingscode)**

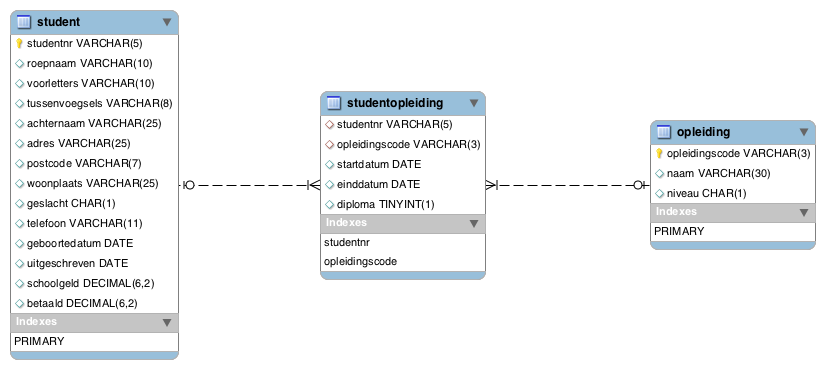
**on delete no action**

**on update no action**

**)**

**engine = InnoDB;**

Tinyint(1) is de database-variant voor een booean in Java. Dat betekent dat het veld diploma alleen de waarde 0 (=false) of 1 (=true) kan hebben. Daarachter staat default 0. Dat betekent dat de beginwaarde altijd 0 (=false) is, omdat we ervan uit gaan dat een student nog geen diploma van de opleiding heeft als hij aan de opleiding begint.

De structuur van de database ziet er dan als volgt uit:

Je ziet hierboven drie tabellen: student, studentopleiding en opleiding. Daarin staan de velden die in de tabel gebruikt worden. Aan het sleuteltje voor een veld kun je de primary key herkennen. Bij studentopleiding worden de twee vreemde sleutels onder indexes genoemd. Deze twee vreemde sleutels worden ook wel een samengestelde sleutel genoemd.

De relaties tussen de tabellen worden in de afbeelding weergegeven door de stippellijnen. In de tabel student moet de student (lees: de sleutel van de student) éėn keer voorkomen, in de tabel studentopleiding komt de sleutel van de student nul keer (de student volgt geen opleiding), éėn keer (de student volgt een opleiding) of meerdere keren voor (de student volgt of heeft meerdere opleidingen gevolgd).

Hetzelfde geldt voor de relatie tussen opleiding en studentopleiding. In de tabel opleiding moet de sleutel van de opleiding éėn keer voorkomen en in de tabel studentopleiding komt de sleutel van de opleiding nul keer (geen student volgt de opleiding), éėn keer voor (een student volgt de opleiding) of meerdere keren voor (Meerdere studenten volgen de opleiding).

De tussentabel studentopleiding maakt het mogelijk dat een student meerdere opleidingen volgt of heeft gevolgd en maakt het mogelijk dat er bij een opleiding meerdere studenten ingeschreven staan of hebben gestaan. De tabel studentopleiding maakt een meer op meer relatie mogelijk terwijl in een relationele database eigenlijk alleen een éėn op veel relatie mogelijk is.

De notatie van Crow:

|  |  |
| --- | --- |
|  | is toegekend aan |
|  | komt nul, een of meerdere keren voor |
|  | komt een of meerdere keren voor |
|  | komt precies een keer voor |
|  | komt nul of een keer voor |

Opdracht 9.4

Zorg ervoor dat de tabel studentopleiding de volgende inhoud heeft (de einddatum blijft leeg:

* 1111, AO, 1-8-2015
* 2222, MV, 1-8-2015
* 3333, DT3, 1-8-2015
* 4444, DT2, 1-8-2015
* 5555, MBI, 1-8-2015
* 6666, IB, 1-8-2015

Als je nu wilt opzoeken, wie welke opleiding volgt, dan doe je dat als volgt:

select roepnaam, tussenvoegsels, achternaam, opleiding.naam from student

inner join studentopleiding on studentopleiding.studentnr = student.studentnr

inner join opleiding on opleiding.opleidingscode = studentopleiding.opleidingscode

where achternaam = "Tol" and tussenvoegsels = 'van der' and roepnaam = 'Berend';

Achter select staan de namen van velden die je wilt zien, achter from staat de naam van de tabel waarvan de meeste velden achter select staan. Daarachter wordt met inner join de twee relaties aangegeven. Eerst staat er de naam van een tabel die nog niet is genoemd achter from of inner join en achter on staat de relatie, gezien vanuit de tabel die voor on is genoemd. Achter where wordt de voorwaarde genoemd waaraan de gegevens moeten voldoen.

Opdracht 9.5

Voer de bovenstaande query uit en zie wat het resultaat is.

Aan de naam inner join kun je afleiden dat er ook een outer join bestaat. Die bestaat inderdaad, maar heet in SQL een left of een right join. Bij een inner join zien we alleen de studenten die een opleiding volgen. Als we alle studenten willen zien, dus ook die geen opleiding volgen, dan gebruiken we de outer join:

select studentnr, roepnaam, tussenvoegsels, achternaam, opleiding.naam from student

left join studentopleiding using(studentnr)

left join opleiding using (opleidingscode);

Opdracht 9.6

Voer de bovenstaande query uit en zie wat het resultaat is.

Opdracht 9.7

Welke opleiding volgt Berend van der Tol? Geef de SQL-opdracht.

Opdracht 9.8

Wie volgt / volgen de opleiding ApplicatieOntwikkeling?

Opdracht 9.9

Welke studenten volgen een opleiding op niveau 4?

Opdracht 9.10

Maak een lijst van de niveaus met daarachter het aantal studenten die een opleiding op dat niveau volgen.

Vaktaal

* **Sleutel of eerste sleutel (Primary key)**: Het veld waaraan een record in de tabel wordt herkend.
* **Vreemde sleutel (Foreign key):** Een veld waarin een sleutel afkomstig uit een andere tabel staat ter verwijzing naar de inhoud van het record in die andere tabel.
* **refererende sleutel**: zie Vreemde sleutel
* **Samengestelde sleute**l: Twee of meerdere refererende sleutels in een tabel die in combinatie uniek zijn.
* **Relatie**: De verwijzing van de ene naar de andere tabel