

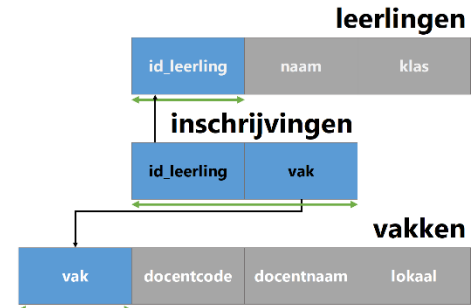
MYSQL DATABASE MAKEN

Inleiding

Tijdens dit practicum maak je een database in MySQL. We doen dit op basis van de casus *examentrainingen*:

Op een school worden examentrainingen aangeboden aan examenklassen. Elke leerling moet drie trainingen kiezen uit het aanbod. Een training hoort bij een bepaald vak en wordt gegeven door één docent. Voor elk vak is er één training die eenmalig wordt gegeven.

Elke docent geeft maar één vak en heeft een unieke docentcode en een eigen lokaal waarin niemand anders lesgeeft.



Voor deze casus is al een strokendiagram gemaakt met daarin tabellen, primaire sleutels en referentiesleutels. Hoe vertalen we dit ontwerp (het strokendiagram) naar een echte database?

Een nieuwe database maken en tabellen toevoegen

1. Klik op de startpagina van de omgeving om *phpMyAdmin* te openen.
2. Klik in het menu op *Databases* en vul onder **Database aanmaken** de naam van de database in: in ons geval *examentrainingen*. Klik vervolgens op *Aanmaken*.
3. Er opent zich een venster waar je tabellen kunt maken. Vul bij **Naam** *leerlingen* in en bij **Aantal kolommen** 3 (corresponderend met het aantal attributen in ons strokendiagram voor de tabel *leerlingen*). Klik vervolgens op *Starten*.
4. We kunnen nu per attribuut aangeven wat de eigenschappen zijn en aan welke eisen het voldoet. Vul de volgende waarden in bij de kolommen:
 - a. **Naam** *id_leerling*, daaronder *naam* en voor het derde attribuut *klas*;
 - b. **Type** (het formaat van de data) *INT* (voor *id_leerling*), *TEXT* (*naam*) en *VARCHAR* (*klas*);
 - c. **Lengte/Waarden** Voor sommige datatypes is het nodig om een maximale lengte te specificeren. Alleen voor *klas* (*VARCHAR*) vullen we 4 in;
 - d. (We slaan nu een aantal kolommen over die we onveranderd laten;)
 - e. **Index** alleen voor *id_leerling* vullen we hier *primary* in om aan te geven dat dit de primaire sleutel van de tabel is. Er opent een venster: klik simpelweg op *Starten*.
 - f. **A_I** staat voor *auto-increment* (automatisch verhogen). Onze primaire sleutel krijgt door te vinken netjes een nieuwe unieke waarde als een nieuw record wordt toegevoegd;

5. Klik op *SQL-voorbeeld*. We lezen:

```
CREATE TABLE `examentrainingen`.`leerlingen` (  
  `id_leerling` INT NOT NULL AUTO_INCREMENT , `naam` TEXT NOT NULL ,  
  `klas` VARCHAR(4) NOT NULL , PRIMARY KEY (`id_leerling`)  
) ENGINE = InnoDB;
```

Dit is de SQL-code waarmee de tabel wordt gegenereerd. (Die zou je dus ook kunnen intypen.)

6. Sluit het *SQL-voorbeeld* en klik nu op *Opslaan*. Links in het menu is nu de tabel *leerlingen* verschenen als onderdeel van de database *examentrainingen*.
7. Klik, helemaal links bij *examentrainingen*, op *Nieuw* om een tabel aan de database *examentrainingen* toe te voegen. Voeg nu zelf de tabel *vakken* toe. Gebruik het type *TEKST* voor *docentnaam* en *VARCHAR* voor *vak* (lengte 16), *docentcode* (lengte 3) en voor *lokaal* (lengte 5). **HINT** vergeet de instellingen voor de primaire sleutel niet (waarom is hier geen *auto-increment* mogelijk?).

Relaties tussen tabellen definiëren en bekijken

De tabel *inschrijvingen* wijkt enigszins af van de twee tabellen die we al hebben gemaakt: er is een gecombineerde sleutel. Alleen de combinatie van *id_leerling* en *vak* is uniek, want:

- Een leerling moet drie trainingen kiezen en dus zal *id_leerling* vaker voorkomen;
- meerdere leerlingen kiezen dezelfde training en dus zal *vak* vaker voorkomen;
- leerlingen mogen maar één keer voor een bepaald vak kiezen en dus moet worden voorkomen dat de combinatie *id_leerling* en *vak* meer dan één keer in de database voorkomt.

8. Maak een nieuwe tabel met de naam *inschrijvingen* en twee attributen. Zorg dat alle attributen hetzelfde datatype en dezelfde lengte krijgen als in de twee tabellen waarnaar ze verwijzen. Je hoeft alleen de kolommen **Naam**, **Type** en **Lengte** te vullen. Daarna klik je op *Opslaan*.

9. Je hebt nu een venster bereikt zoals hiernaast. Hierin maken we de samengestelde sleutel. Vink daarvoor de attributen *id_leerling* en *vak* aan en klik vervolgens onder de tabel op *Primaire sleutel* zodat net als in het screenshot hiernaast sleuteltjes verschijnen achter *id_leerling* en *vak*.

#	Naam	Type	Collatie	Attributen	Leeg	Standaardwaarde	Opmerkingen	Extra	Actie
<input type="checkbox"/>	1 id_leerling	int			Nee	Geen			Veranderen
<input type="checkbox"/>	2 vak	varchar(16)	utf8mb4_0900_ai_ci		Nee	Geen			Veranderen

Met geselecteerde: Verkennen Veranderen Verwijderen Primaire sleutel

Het is nu tijd om de relaties tussen de tabellen aan te geven. In de tabel *inschrijvingen* zijn *id_leerling* en *vak* referentiesleutels: ze verwijzen (refereren) naar andere tabellen. Door de relaties te maken, controleert MySQL de referentiële integriteit. Dit voorkomt bijvoorbeeld dat er in de tabel *inschrijvingen* een leerling (*id_leerling*) voorkomt die helemaal niet bestaat in de tabel *leerlingen*.

10. Klik bovenaan op *Relatieoverzicht*. Je komt nu in een venster waar je *Beperkingen voor vreemde sleutels* (referentiesleutels) kunt aangeven.

11. Selecteer (zoals hiernaast weergegeven) onder **Kolom** het attribuut *id_leerling* en koppel deze vervolgens aan de tabel (**Tabel**) *leerlingen* via het attribuut (**Kolom**) *id_leerling*. Klik daarna op *Opslaan*.

Kolom

Vreemde sleutel (INNODB)

Database

Tabel

Kolom

id_leerling

examentraining

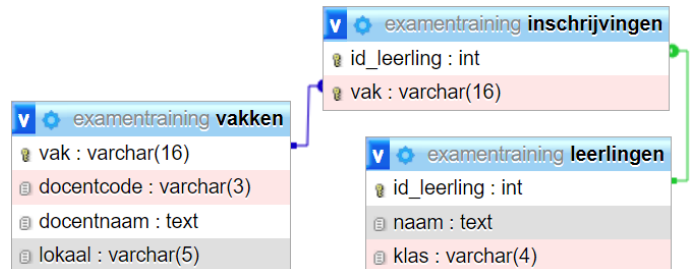
leerlingen

id_leerling

+ Kolom toevoegen

12. Registreer op vergelijkbare manier de relatie tussen de tabel *inschrijvingen* en *vakken*.

13. Klik bovenaan of links op *examentrainingen* om de startpagina van de database te bereiken. Klik daar rechts op *Ontwerper*. Je krijgt dan een beeld zoals hiernaast (zelf schuiven).



14. Controleer of bij jou de primaire sleutels goed staan. Check ook richting van de referenties.

De database vullen

Nu de structuur is vastgelegd, kunnen we de database vullen.

15. Open *H3O02 examentrainingen.xlsx* en bekijk de tabelinhoud.

16. Selecteer in *PHPMyAdmin* de tabel *leerlingen* (links of via de startpagina van de database). Klik vervolgens op *Invoegen*.

17. Maak vijf records aan voor de tabel *leerlingen* op basis van de gegevens in het Excel-document. Het venster wijst zich vanzelf.

18. Selecteer de tabel *leerlingen* en klik op *Verkennen*. Als het gelukt is zie je een beeld vergelijkbaar met het screenshot hiernaast.

19. Vul de tabel *vakken* op basis van de gegevens in het Excel-document.

id_leerling	naam	klas
1	Alice Pieters	H5A
2	Bob Bouwma	H5A
3	Cindy Young	H5B
4	Dirk Nauta	H5A
5	Eve Ning	H5B

Data importeren

Vijf records invullen voor leerlingen is nog wel te doen, vijf vakken wordt al vervelender. Gelukkig kan data worden geïmporteerd. Een veelgebruikt bestandsformaat is CSV. Dit staat voor *Comma-separated values* maar staat algemener voor een bestand met vaste scheidingstekens (zoals “” of ; of tab).

20. Open *H3O02 examentrainingen_inschrijvingen.csv* (niet in Excel maar bijvoorbeeld in *Kladblok!*). Stel vast welke scheidingstekens hier worden gebruikt.
21. Selecteer de tabel *inschrijvingen* en klik op *Importeren*. Selecteer bovenaan ons csv-bestand.
22. Navigeer naar **Indelingsspecifieke opties**: en vul hier de passende gegevens in zoals hiernaast weergegeven. Klik daarna op *Starten*.
23. Bekijk de geïmporteerde data via *Verkennen*. Merk op dat je kunt klikken op attribuutwaarden van de attributen *id_leerling* en *docentcode*. Wat gebeurt er als je er op klikt?

Indeling:

CSV

Opmerking: Als het bestand meerdere tabellen bevat, zullen

Indelingsspecifieke opties:

☐ Werk data bij wanneer dubbele sleutelvel

Kolommen gescheiden door: ;

Kolommen omsloten met:

Kolommen omgezet met wisseltken:

Een database exporteren in MyAdmin

Databases worden niet als gewoon bestand opgeslagen. Je doet er goed aan om je database te exporteren en dat bestand goed te bewaren (b.v. in *Github*). We kijken eerst hoe dit gaat in *PHPMyAdmin*:

24. Navigeer (links of bovenaan) naar de startpagina van de database *examentrainingen*. Klik in het menu op *Exporteren*.
25. Kies in dit venster voor (de reeds ingestelde) waarden *Snel* (**Export methode:**) en *SQL* (**Indeling:**) en klik op *Starten*. Sla het bestand op.
26. Open het opgeslagen sql-bestand en bekijk het overzicht aan SQL-commando's dat wordt gebruikt om jouw database te genereren.

Het kan handig zijn om een iets uitgebreider sql-bestand te maken.

27. Ga nogmaals naar *Exporteren* maar kies nu niet voor *Snel* maar voor *Uitgebreid*. Bekijk eens rustig alle opties maar zorg in ieder geval dat *Bekijk output als tekst* geselecteerd is en dat er (onder **Object-aanmaakopties**) een vinkje staat bij *CREATE DATABASE* en bij *DROP TABLE*. Klik op *Starten* om de uitvoer te bekijken.

Het voordeel van deze toevoegingen is dat je een database sneller kunt herstellen. Je hoeft bijvoorbeeld niet eerst een oude database te verwijderen of een lege database aan te maken.

Werken met onze eigen omgeving

Onze werkomgeving heeft een tweede functie om een database te exporteren.

28. Ga naar de startpagina van de omgeving en klik op *Exporteren*.
29. Ga nu naar Visual Studio Code en navigeer links naar de map *databases*. Stel vast dat hier nu een nieuw bestand *examentrainingen.sql* is verschenen.
30. (Optioneel) *Push* dit bestand (en andere updates) naar je eigen *Github-repository* (als je die hebt aangemaakt) om de database voor de toekomst veilig te stellen.
31. Open de *SQL-querier* en selecteer de database *examentrainingen*. Maak een query die een overzicht geeft van de namen van leerlingen die een training voor Engels hebben gekozen met daarbij de naam van de docent van de training.

Database beheer

Er zijn 3 databases beschikbaar in MySQL:

- top_2000_v1 (standaard)
- (standaard)
- (standaard)

Exporteer de databases naar de databases map.

Exporteren Importeren phpMyAdmin

Herstel standaard databases