­­­­­mysql -u root (basiscommando voor mariadb)

SHOW DATABASES; (keywords in CAPS)

USE forta; (database veranderd van none naar forta)

SHOW TABLES; (toont verschillende tabellen in forta database)

SHOW COMUMS FROM products;

SELECT \* FROM products; ( \* = alles )

SELECT prod\_id, prod\_price FROM products; (voor selectie van gekozen objecten)

­­­­­mysql -u root -p (password ) (-h is andere server : via ip adress of url. Poort = -P note; hoofdletter)

-Nieuwe database aanmaken

CREATE DATABASE demo;

USE demo;

SHOW TABLES; (empty set – geen tabellen)

* Schema maken wat je allemaal wil bijhouden
  + PK EAN (Primary Key – enigste die uniek is) int(13)
  + Name Char(32) – Aantal characters
  + Calories Int (integers, nummers)
  + Weight Double
  + Price Double

CREATE TABLE food (

ean int(13),

name char(32),

calories double,

price double,

PRIMARY KEY (ean),

);

\*\*/CREATE TABLE name (

columnname Datatype,

PRIMARY KEY (comumn\_name\_xyz)

);

/\*\*

SELECT \* FROM food;

-empty set

INSERT INTO food VALUES (1234567890123, ‘Pizza’, 700, 3.5);

INSERT INTO food (ean, name, price) VALUES (12300000000, ‘Frietjes’, 3.5) – Gewicht wordt weggelaten dus duiden we aan welke rows we aanpassen – Dit wordt NULL

INSERT INTO food (price, ean, name) VALUES (10.5, 11234123, ‘Pitta’);

SELECT food WHERE calories is NULL; - alle resultaten calorieën 0

SELECT \* FROM food WHERE calories IS NOT NULL;

SHOW COLUMN FROM food

Create / read / update / delete – CRUD

Vb. Webshop – Nieuwe bestelling plaatsen (klant)

Klant (enkel uitlezen – Read)

Stock (creëren nieuw product stock aanpassen)

Werking CHAR() - makkelijker om op te zoeken

Bert…………………… char(32) – de overgebleven characters worden spaties

Natascha……………………. char(32)

Jo………………………………. char(32)

96 chars

Werking VARCHAR()

VARCHAR(32) 4 Bert 8 Natacsha 2Jo

17 chars

Vb

Werking ENUM: Met ENUM kan je snel toevoegen maar slecht 1 mogelijke waarde selecteren

ENUN (‘Nintendo’, ‘Xbox’, ‘Gamecube’,’Wii’, ‘Playstation’)

1 2 3 4 5

Werking SET: Beperkt aantal items maar meerdere mogelijke waarden selecteren.

8bytes - 00000000 (elk getal komt overeen met een console)

* Zo kunnen meerdere items geselecteerd worden

Git clone

Examenvraag : tabel kennen

|  |  |  |
| --- | --- | --- |
|  | DDL (Data Definition Language)   * structuur | DML (Data Manipulation)   * data |
| C(reate) | CREATE TABLE | INSERT INTO tablename (kolomnaam) VALUES(waarden) |
| R(ead) | SHOW COLUMNS FROM  Alias van SHOW (DESCRIBE) | SELECT \* FROM |
| U(pdate) | ALTER TABLE ADD columnname datatype;  MODIFY columnname datatype;  CHANGE oldname newname datatype; | UPDATE tablename SET name=”Foo” |
| D(elete) | DROP TABLE tablename  DROP COLUMN columnname | DELETE FROM tablename WHERE |

Mysql -u root

SHOW DATABASES;

USE forta;

-/demo/-

CREATE TABLE foo (id INT, name VARCHAR(32), PRIMARY KEY(id));

SHOW COLUMNS FROM foo; of DESCRIBE foo;

ALTER TABLE foo ADD bar ENUM(‘XBOX’, ‘Playstation’,’Wii’);

ALTER TABLE foo CHANGE bar console ENUM(‘XBOX’,’Playstation’,’Wii’);

ALTER TABLE foo ADD bar ENUM(‘XBOX’, ‘Playstation’,’Wii’); - adden voor vb deleten column

ALTER TABLE foo DROP column bar;

SELECT \* FROM foo; of DESCRIBE foo;

INSERT INTO foo (console, id, name) VALUES (1, ‘PUBG’, ‘XBOX’); - fout gemaakt met id

DELETE FROM foo;

INSERT INTO foo (console,id,name) VALUES (‘XBOX’, 100, ‘PUBG’);

SELECT \* FROM foo; of DESCRIBE foo;

INSERT INTO foo (console, id, name) VALUES (‘Wii’,200, ‘Wii Sports’);

UPDATE foo SET console=’Playstation’;

UPDATE foo SET console=’Wii’ WHERE id=200;

SELECT \* FROM foo; of DESCRIBE foo;

DELETE FROM foo WHERE id=200;

SELECT \* FROM foo; of DESCRIBE foo;

DROP TABLE foo;

ALTER TABLE movielist MODIFY COLUMN genre ENUM(‘…’,’…’,’…’,..);

ER – diagram (Entity – Relation)

Vb. Games

Games -

ER Diagram

|  |  |  |  |
| --- | --- | --- | --- |
|  | Conceptual | Logical | Physical |
| Entity | X | X |  |
| Entity Relations | X | X |  |
| Attributes |  | X |  |
| PK |  | X | X |
| Foreign Key |  | X | X |
| Tablenames |  |  | X |
| Columnnames |  |  | X |
| Datatypes |  |  | X |

Entity = Tablenames maar op een ander niveau

Attributes = Columnnames maar op een ander niveau

Foreign Key = Primary Key van een andere table

**Normalisatie :**

* **elimineren van duplicate data**
* **Data integriteit verzekeren (3 soorten anomaliteiten)**
  + **Problemen bij Insertion – vak toevoegen**
  + **Problemen bij Updation – vak toevoegen**
  + **Problemen bij Deletion - vak verwijderen**
    - **Normal Form (normaal vormen ter oplossing van de anomaliteiten)** 
      * **1 NF**
      * **2 NF**
      * **3 NF**
      * **BCNF**

Table : Teachers

Slechte table. Vb.

Zoeken naar leerkracht dat OOP2 geeft.

SELECT name FROM teachers WHERE course1 =”OOP2” OR course2 =”OOP2” OR course3=”OOP2”

1 NF ( First normal form)

* Data in database table (relatie)
  + Relatie
    - Each row contains data about an entity.
    - Columns contain data about a property of the entity
    - All values in each column must be of the same type
    - Each column must have a unique name
    - Orders (volgorde) of row is not important
    - Orders of columns is not important

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| u-nr | name | campus | tel-nr | course1 | course2 | course3 |
| U000123 | Sille | Station | 047712345 | Wechtech | Database | Projectwerk2 |
| U000789 | Nico | Station | 047712345 | OOP1 | OOP2 |  |
| U123456 | Luc | Oostende | 050001122 | Vliegtuigmotoren |  |  |

* + - No two rows may be the same
* Each column contains an atomic value – (kleinste waarde dat je niet meer kan delen)

⏵

Teachers table fixen:

|  |
| --- |
| Teachers |
| U nr |
| Name |
| Campus |
| Telnr |
| Course1 |
| Course2 |
| Course3 |

|  |
| --- |
| **Teacher** |
| u-nr |
| name |
| campus |
| telnr |

|  |
| --- |
| **Course** |
| id |
| name |

|  |  |  |  |
| --- | --- | --- | --- |
| Unr | name | campus | telnr |
| U000123 | Sille | Station | ----- |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Id | name | teacher\_u\_nr |
| 1 | webtech | u000123 |
| 2 | database | u000123 |
| 3 | projectwerk | u000123 |
| 4 | Webtech | u12345 |

Foreign Key

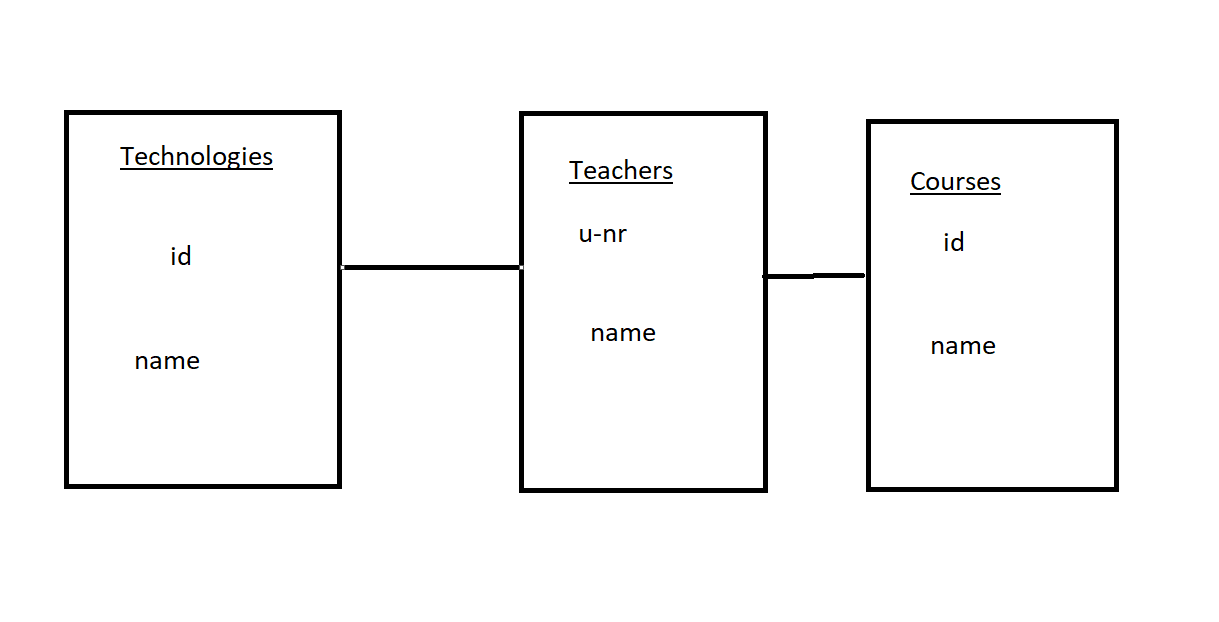
Voorbeeld van duplicate data – veel redundantie

|  |  |  |  |
| --- | --- | --- | --- |
| Teachers | | | |
| u-nr | name | courses | Technologies |
| U00123 | Sille | Webtech | HTML |
| U00123 | Sille | Databases | HTML |
| U00123 | Sille | Webtech | Docker |
| U00123 | Sille | Database | Docker |

|  |
| --- |
| Teachers |
| Unm |
| Name |
| Course |
| technology |

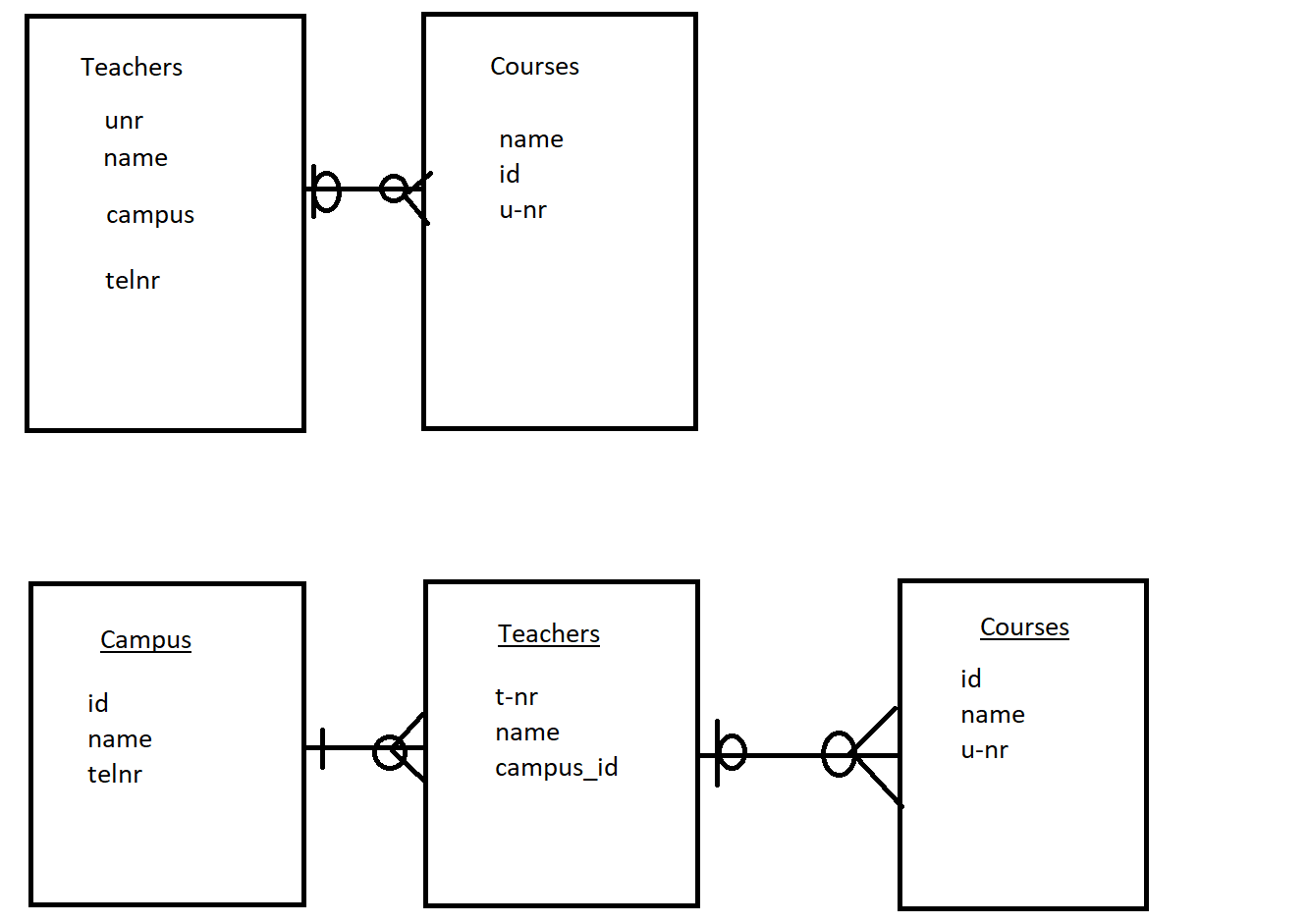
⏷

Oplossing:



2NF

* 1NF
* Alle attributen moeten afhankelijk zijn van hun Primary Key

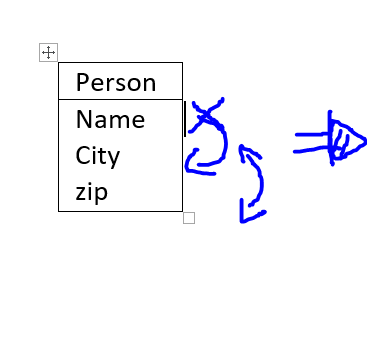
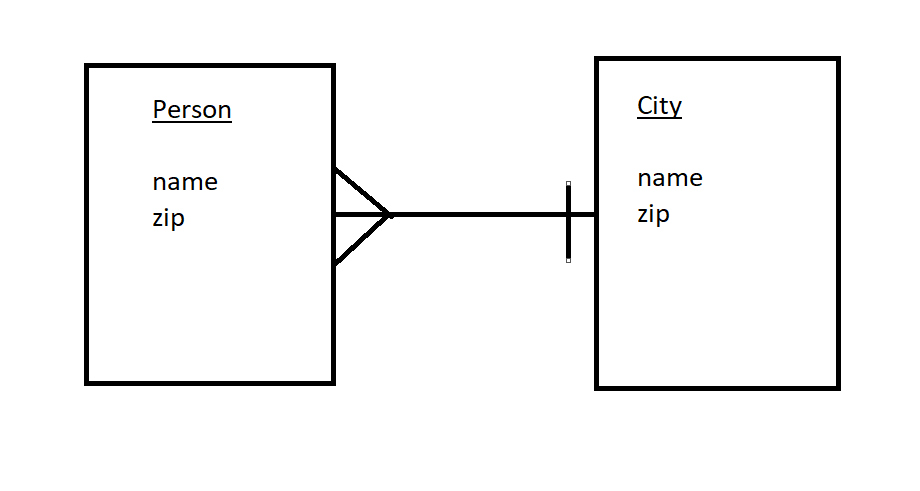


3NF / niet kennen – weten wat het is

* 2NF
* Contains only columns that are non transversely dependent on the primary key

Voldoet niet:

|  |
| --- |
| Person |
| Name  City  zip |

BCNF : Boyce – Cod NF / niet kennen maar weten wat het is

* 3NF
* Each functional dependency (x -> y) x is a super key

SHOW TABLE STATUS; opvragen engine type

SELECT \* FROM person WHERE name = “Michiel”;

SELECT value FROM weight WHERE person\_id=1

SELECT value

FROM weight

JOIN person

WHERE name = ‘Michiel’;

SELECT \* FROM weight JOIN person ON weight.person\_id = person.id WHERE name = ‘Michiel’;

SELECT film.name

FROM films

JOIN user\_film ON film.id = user\_film.fillm\_id

JOIN user ON user.id = user\_fil.user\_id

WHERE user.name = ‘Michiel’