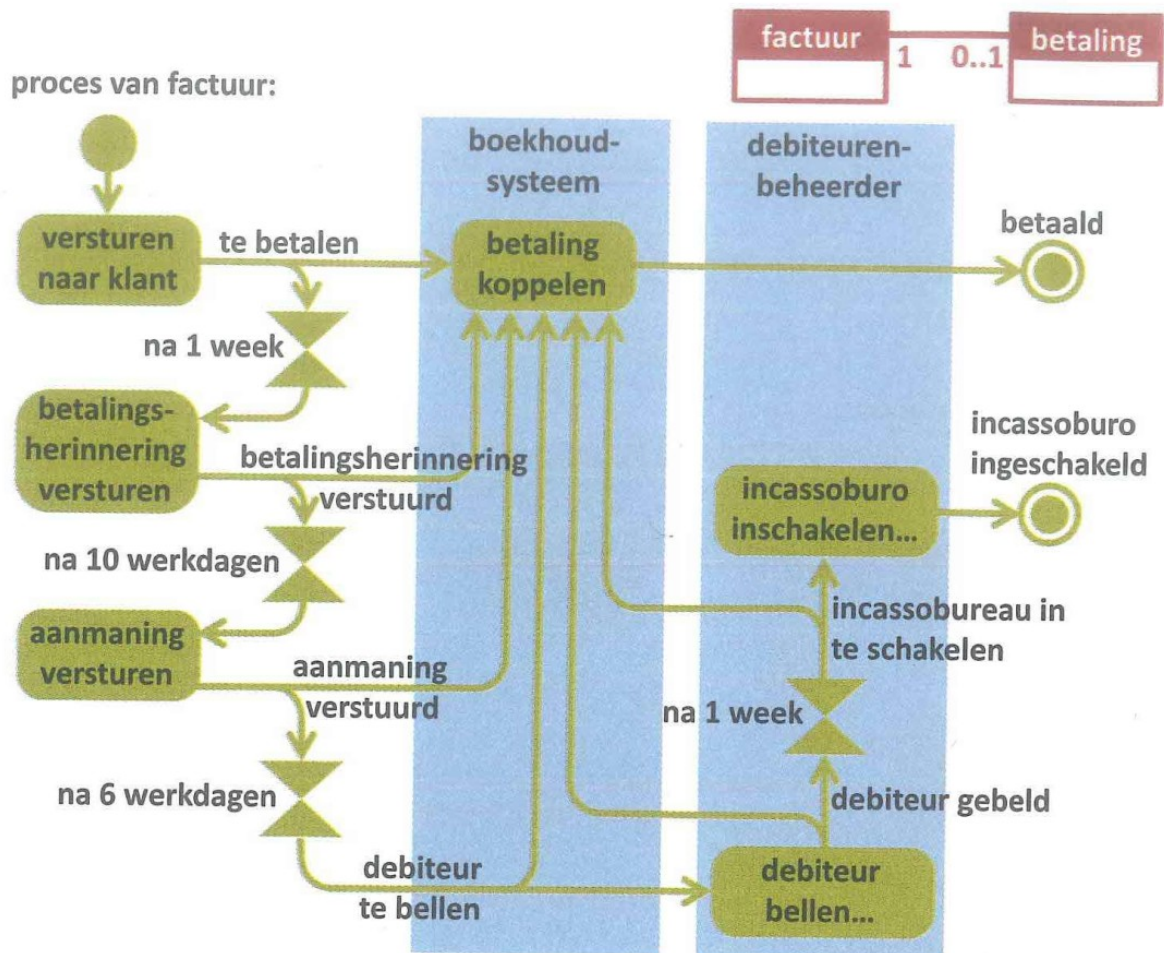
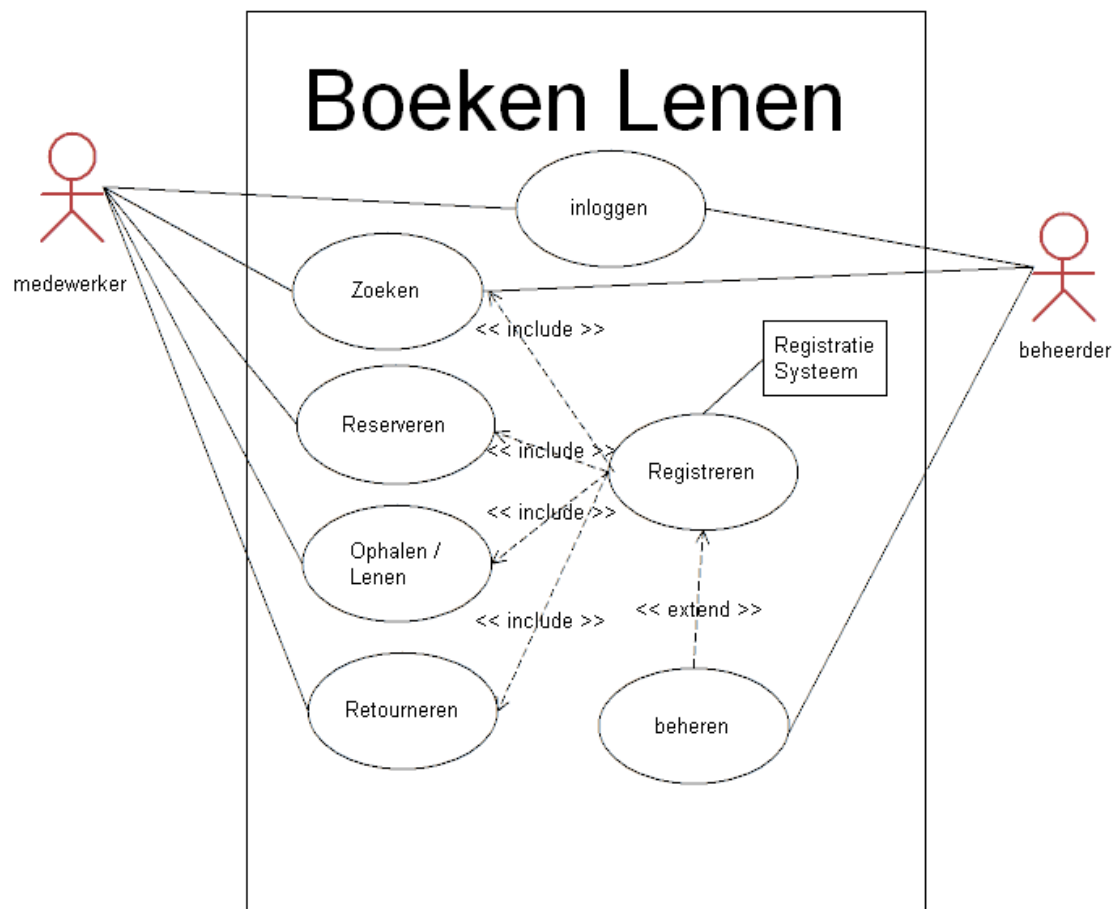
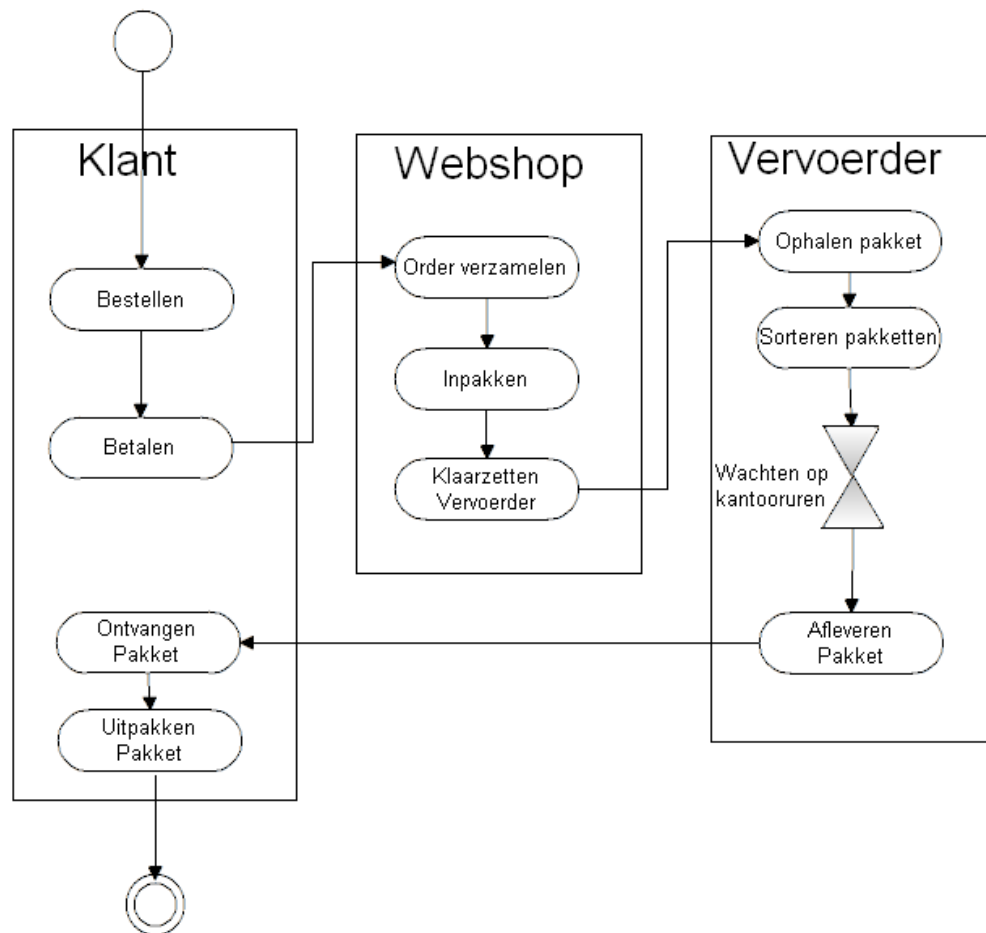
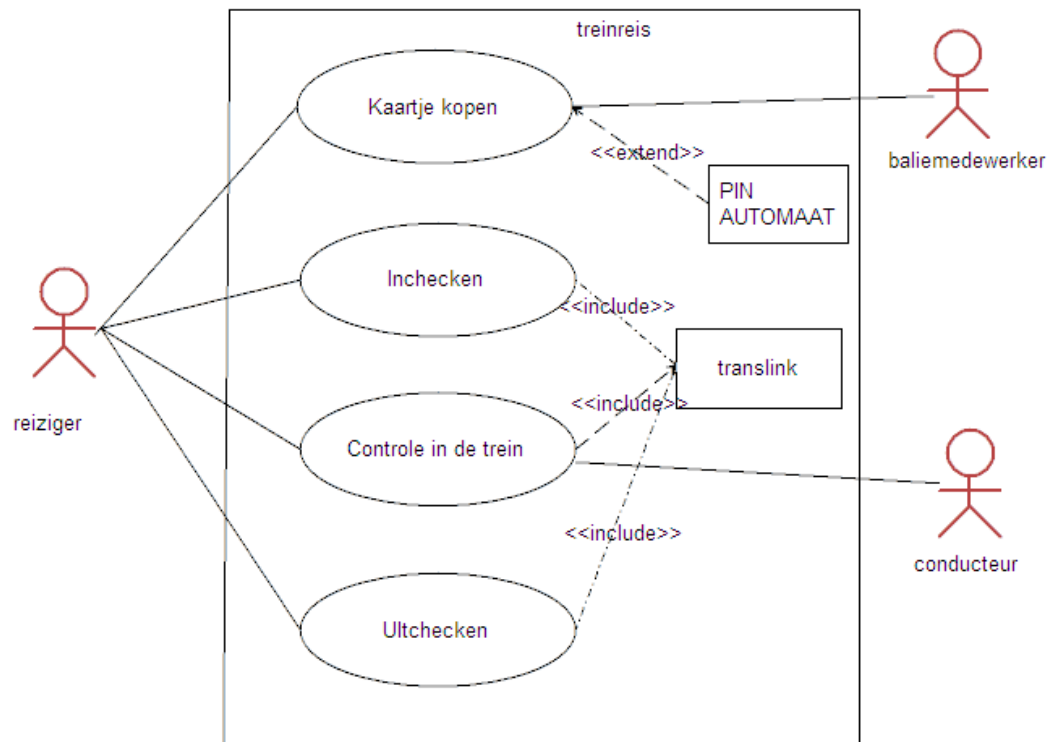


Het activity diagram

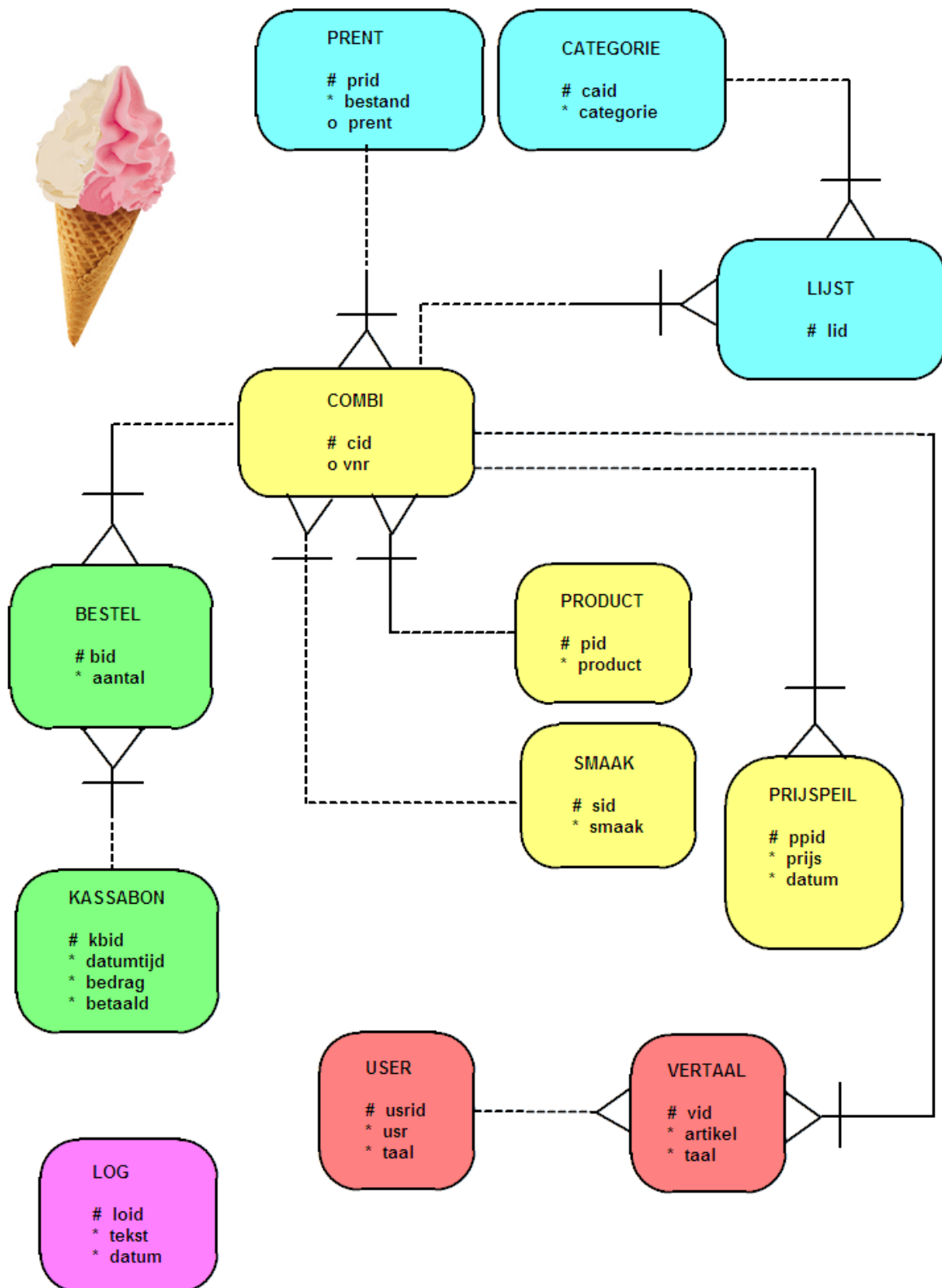




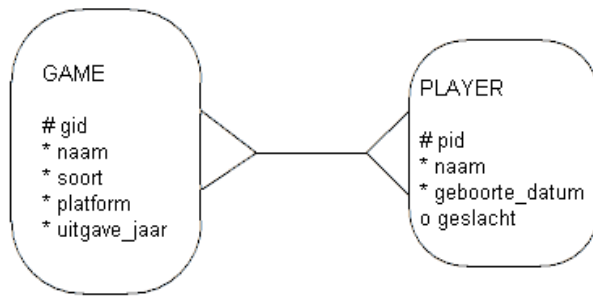




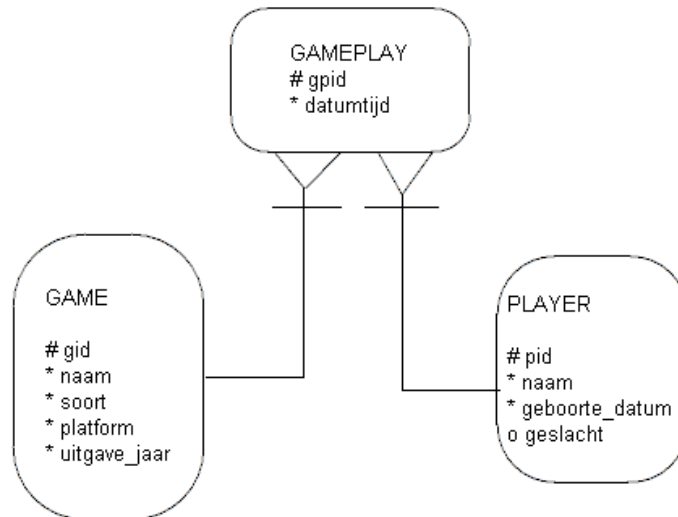
Use case diagram
treinreiziger



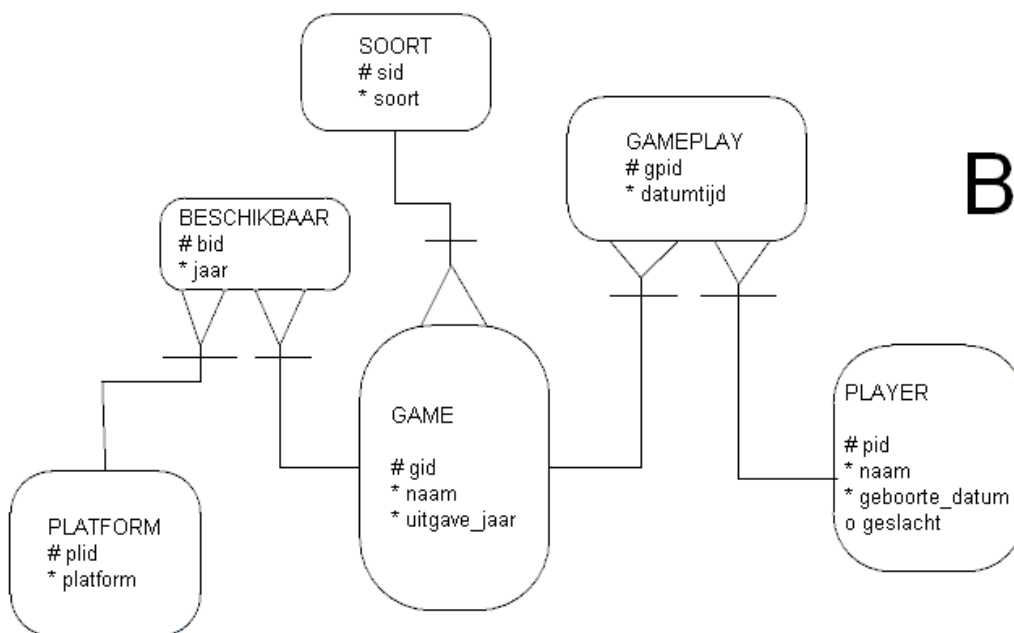
ERD de IJssalon



BUT



BETER



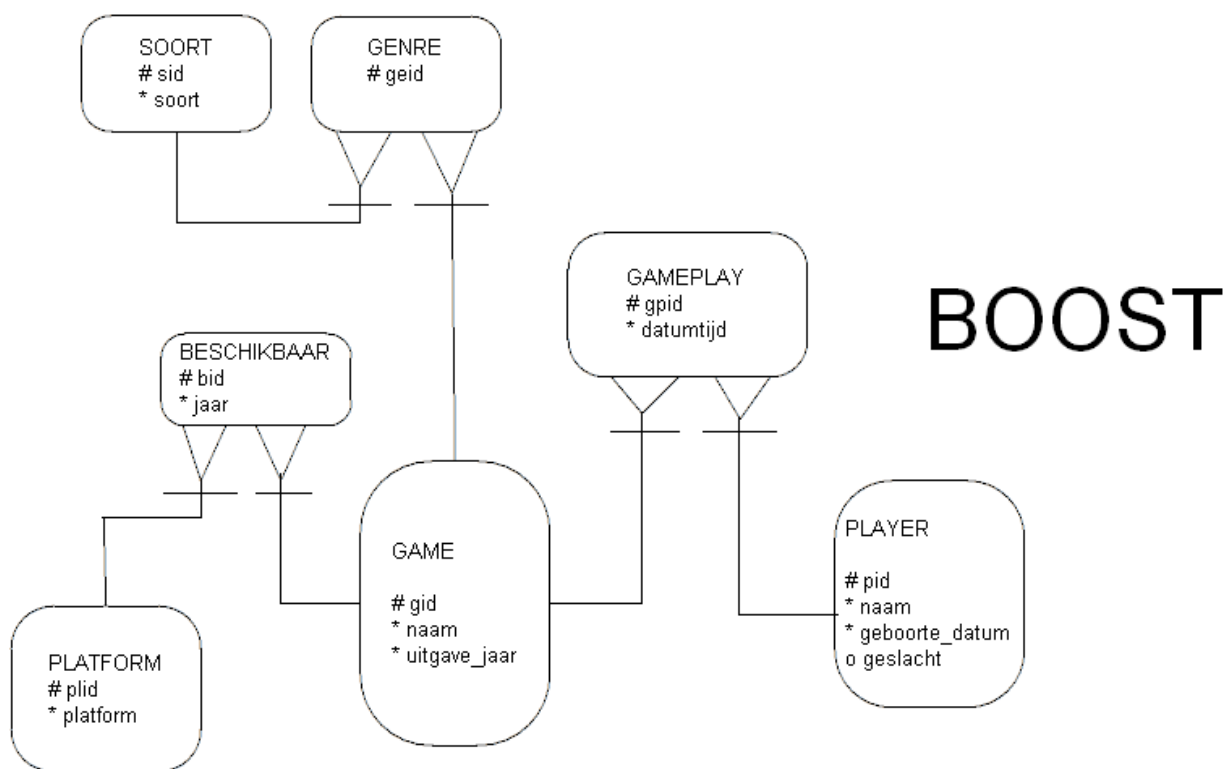
BEST

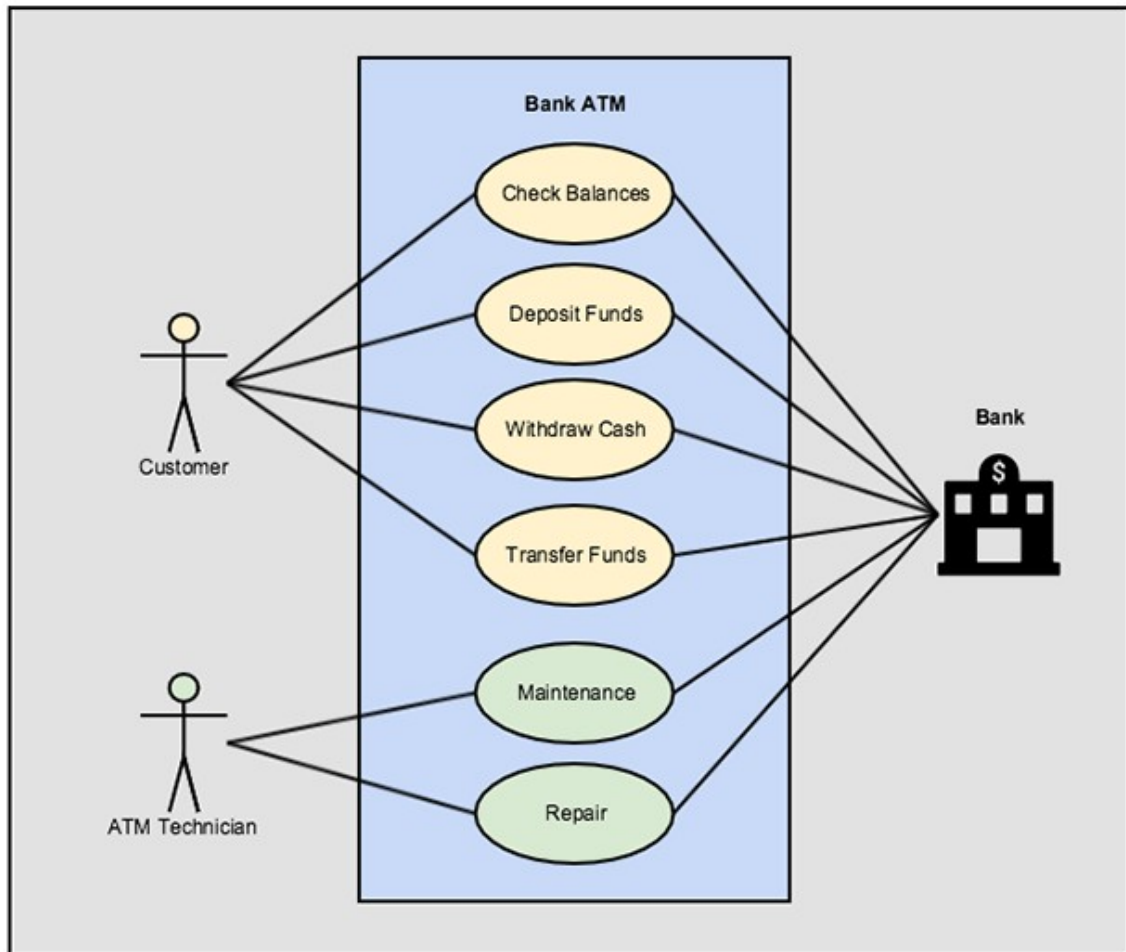
In de BUT versie ontbreekt de tussen entiteit en kun je dus geen lijst maken *wie welke spelletjes speelt*.

In de BETTER versie is de tussen entiteit aanwezig en kun je spelletjes aan spelers koppelen.

De BEST versie is nog beter, want hier kun je spelletjes aan meerdere platforms koppelen.

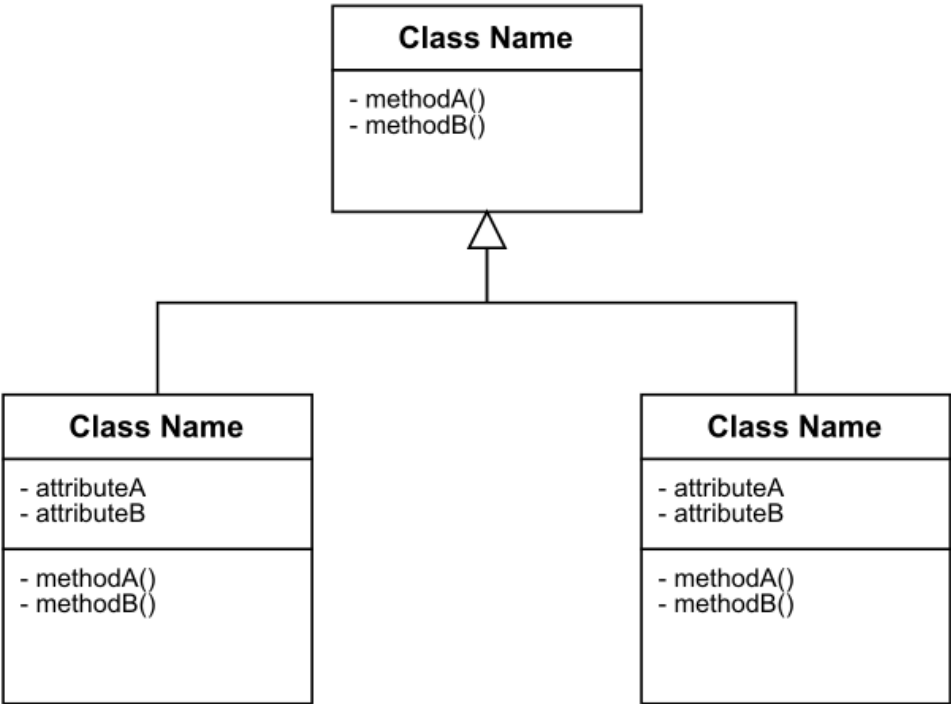
En de BOOST versie is ultiem, hierin kan een spel meerdere genre's (soort) hebben.



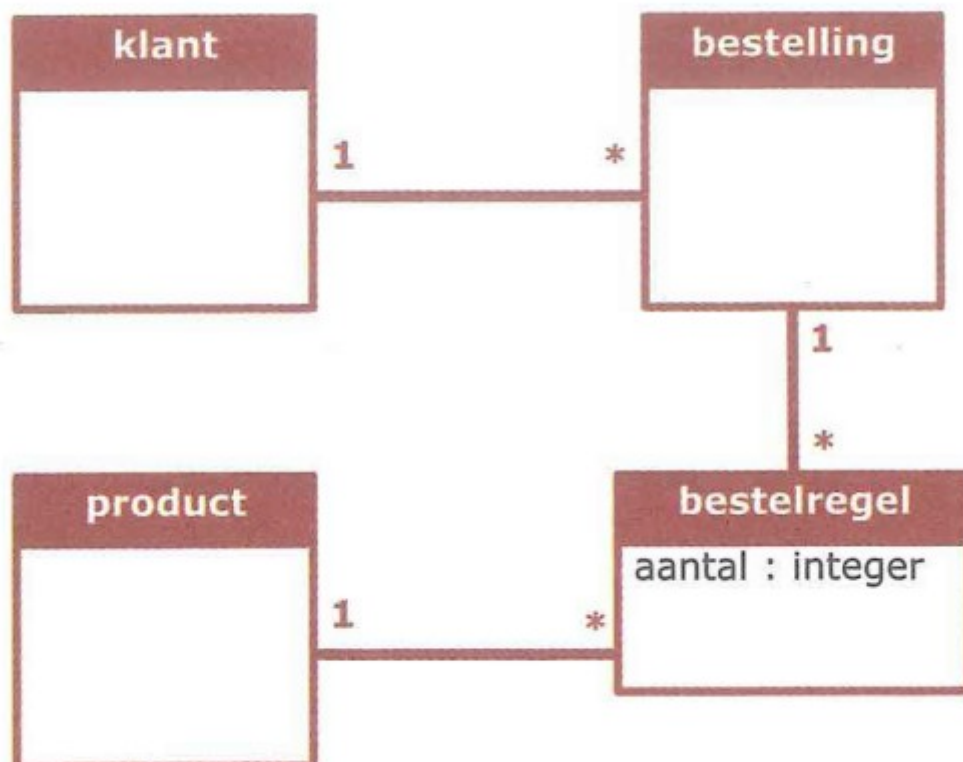
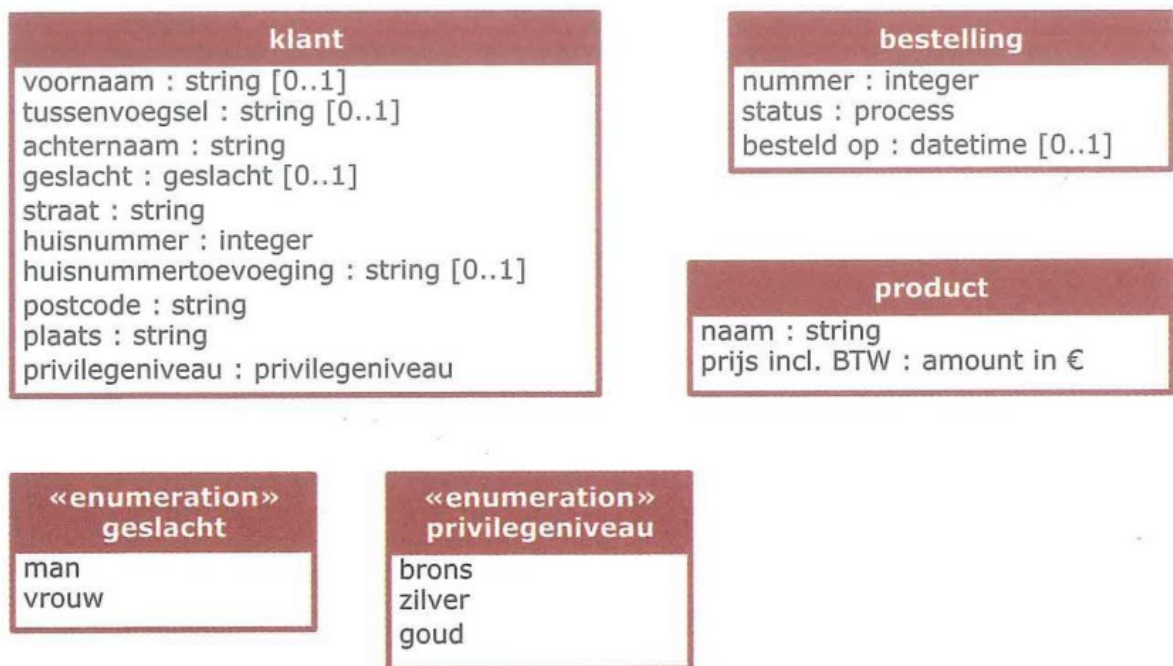


De ATM technician zou denkkelijk aan de andere kant moeten staan. De Bank had ook als system actor binnen het hok kunnen staan.

Class Diagram



Class Diagram



rectangle

left : integer
top : integer
right : integer
bottom : integer

calculateSurface() : integer
move(deltaX : integer, deltaY : integer) : void

Software Engineering



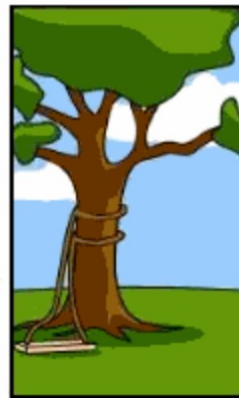
How the customer explained it



How the Project Leader understood it



How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



How the project was documented



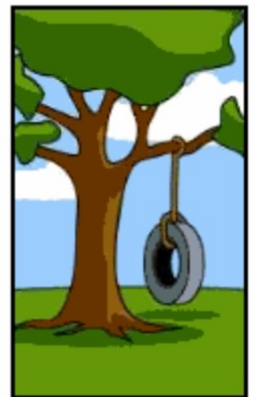
What operations installed



How the customer was billed

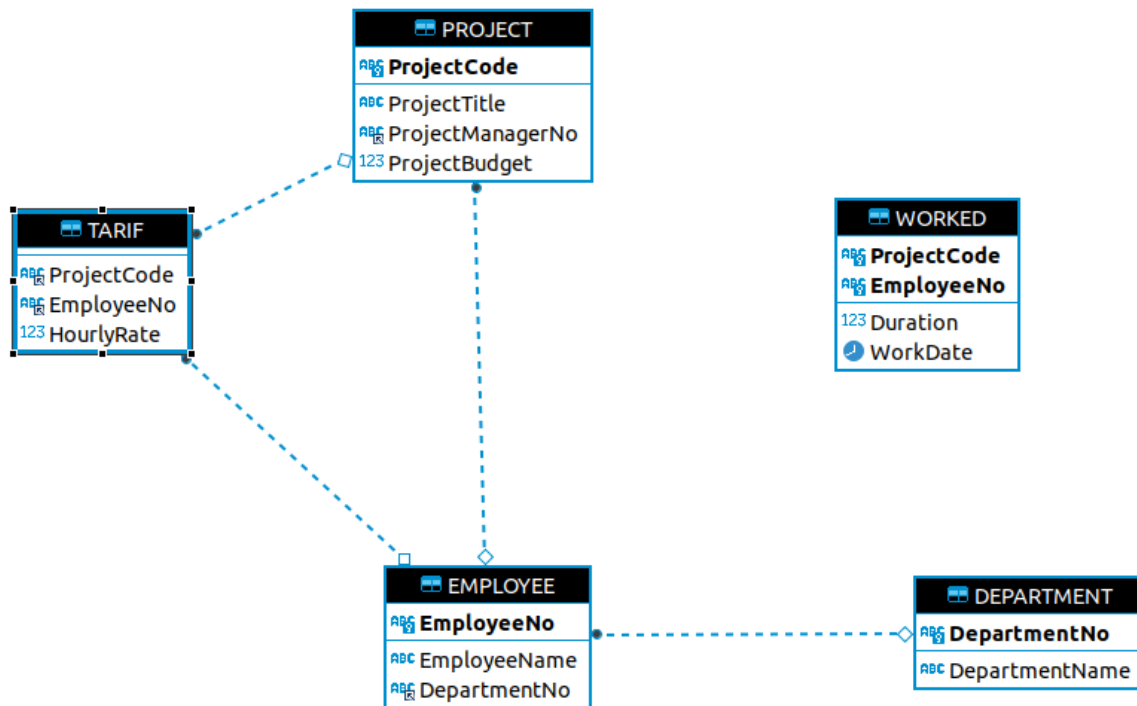


How it was supported

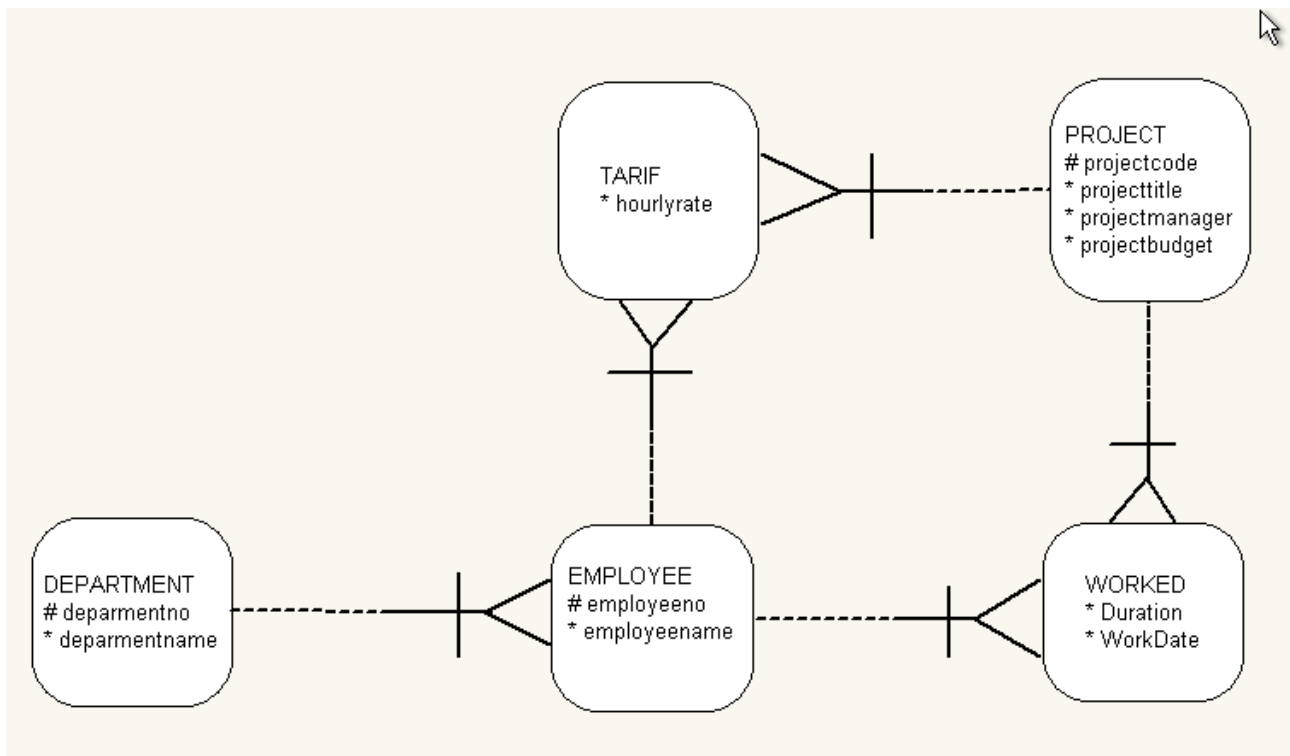


What the customer really needed

Fysiek model



In dit fysieke model ontbreken de relaties nog met de WORKED tabel.



Dit is het Oracle ERD met de relaties. Let op de stippellijnen.

Een Employee hoeft niet perse in WORKED of TARIF te staan. Maar een Employee moet altijd bestaan als deze in TARIF of WORKED wordt ingevoerd.

Als je naar de WORKED entiteit kijkt, zie dan het verschil met de WORKED tabel. Deze laatste heeft twee extra velden die via die **bar** in de ERD vermeld staan.

Maak de database games aan:

```
CREATE DATABASE IF NOT EXISTS `games` /*!40100 DEFAULT CHARACTER SET utf8mb4 */;
```

Gebruik de database:

```
USE games;
```

Creeer de tabellen:

```
DROP TABLE IF EXISTS `game`;
```

```
CREATE TABLE `game` (  
  `g_id` int(11) NOT NULL AUTO_INCREMENT,  
  `g_naam` varchar(50) DEFAULT "",  
  `g_soort` varchar(20) DEFAULT "",  
  `g_platform` varchar(20) DEFAULT 'PC',  
  `g_uitgavejaar` char(4) DEFAULT NULL,  
  PRIMARY KEY (`g_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
DROP TABLE IF EXISTS `player`;
```

```
CREATE TABLE `player` (  
  `p_id` int(11) NOT NULL AUTO_INCREMENT,  
  `p_naam` varchar(50) DEFAULT "",  
  `p_gebdatum` date,  
  `p_geslacht` char(1) DEFAULT 'O',  
  PRIMARY KEY (`p_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
DROP TABLE IF EXISTS `gameplay`;
```

```
CREATE TABLE `gameplay` (  
  `gp_id` int(11) NOT NULL AUTO_INCREMENT,  
  `g_id` int(11) NOT NULL,  
  `p_id` int(11) NOT NULL,  
  `gp_datumtijd` datetime,  
  PRIMARY KEY (`gp_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

Voer de onderstaande gegevens in:

```
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Dune 2000', 'Strategie', '1998' );  
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Polybius', 'onbekend', '1981' );  
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Doom', 'FPS', '1993' );  
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Half Life', 'FPS', '1998' );
```

```
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Half Life II', 'FPS', '2004' );  
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Black Mesa', 'FPS', '2015' );
```

```
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Xonotic', 'FPS', '2017' );  
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Unreal Tournament 2004', 'FPS', '2004' );  
INSERT INTO game ( g_naam, g_soort, g_uitgavejaar )  
VALUES('Warzone 2100', 'Stategie', '1999' );
```

```
INSERT INTO game ( g_naam, g_soort, g_platform, g_uitgavejaar )  
VALUES('Pikmin', 'Strategie', 'Nintendo Gamecube', '2001' );  
INSERT INTO game ( g_naam, g_soort, g_platform, g_uitgavejaar )  
VALUES('Lemmings', 'Strategie', 'Amiga', '1991' );  
INSERT INTO game ( g_naam, g_soort, g_platform, g_uitgavejaar )  
VALUES('Pacman', 'Arcade', 'Arcademachine', '1980' );
```

-- Datums voer je altijd in ISO formaat in: YYYY-MM-DD

```
INSERT INTO player ( p_naam, p_gebdatum, p_geslacht )  
VALUES('lp13', '1962-12-14', 'M' );
```

```
SELECT * FROM game;
```

```
SELECT * FROM player;
```


Voer de gegevens in de gameplay tabel middels een query!

```
INSERT INTO gameplay ( g_id, p_id, gp_datumtijd )  
SELECT g_id, p_id, now()  
FROM game, player WHERE NOT (g_naam LIKE '%Pik%') AND NOT (g_naam LIKE '%Poly%')  
AND NOT EXISTS ( SELECT * FROM gameplay );
```

```
SELECT * FROM gameplay;
```

```
– TRUNCATE TABLE gameplay;  
– TRUNCATE TABLE game;
```

```
SELECT *  
FROM game  
NATURAL JOIN gameplay  
NATURAL JOIN player;
```

```
– Haal de gegevens over van de games database naar  
– games_boost database.  
USE games_boost;
```

```
INSERT INTO game ( g_naam, g_soort, g_platform,  
g_uitgavejaar )  
SELECT g_naam, g_soort, g_platform, g_uitgavejaar  
FROM games.game;
```

```
getal1 = random(100)  
getal2 = random(100)
```

```
IF (getal2 > getal1):  
    getal3 = getal2  
    getal2 = getal1  
    getal1 = getal3
```

```
antwoord = getal1 - getal2
```

MAAK DE OPDRACHT VAN DE TREINREIZIGER

TOT VRIJDAG!