

Uitwerking Practicum Fietsverhuur

Practicum stap 1: Entiteiten, attributen en identifiers.

Lees het stukje hieronder en maak daarna de vragen.

Ik ben de eigenaar van een klein fietsenverhuurbedrijf. We kunnen meer dan 3000 fietsen uitlenen. Iedere fiets heeft een eigen identificatienummer groot op het spatbord of frame ingegraveerd. Verder noteren we het framenummer in verband met mogelijke diefstal. We lenen stadsfietsen uit maar ook racefietsen en mountainbikes enzovoort. Van veel van deze soorten fietsen hebben we ook een e-bike uitvoering. Verder hebben we een heren- en een damesmodel van de meeste fietsen. Van ieder model hebben we diverse wielmaten staan waaronder ook kindermaten. De maten lopen van 12 inch tot en met 60 inch. Zo is er voor iedereen wat.

We kopen onze fietsen bij verschillende leveranciers. Van iedere leverancier noteren we zijn bedrijfsnaam, kamer van koophandel nummer, woonplaats, adres en telefoonnummer.

Ja, van vele fietsen hebben we meerdere exemplaren staan. Vooral van de meest verhuurde fietsen.

- a) Kopieer bovenstaande tekst met vragen naar een document dat je "Antwoorden practicum fietsverhuur" noemt.
- b) Markeer of noteer alle zelfstandige naamwoorden uit bovenstaand stukje.
- c) Kies uit bovenstaand stukje de attributen (eigenschappen) en noteer erachter waar het een eigenschap van is. Op die manier vindt je de entiteiten.
- d) Noteer de namen van de entiteiten in enkelvoud.
- e) Noteer bij iedere entiteit de attributen die daarbij horen.
- f) Is iedere voorbeeld van een entiteit uniek? Noteer bij iedere entiteit welke eigenschap als primary UID (unique identifier) kan fungeren. Noteer indien beschikbaar ook één of meer secondary UID(s).
- g) Teken de entiteiten in een ERD (Entity Relationship Diagram) compleet met attributen en de tekens ervoor of ze de primary UID #, secondary UID (#), mandatory * (verplicht) of optional o (niet verplicht) zijn. Je hoeft nog geen verbindinglijnen tussen de entiteiten te trekken.
- h) Controleer of je model zo is gemaakt dat als de waarde van één eigenschap zou veranderen, dat je dit dan slechts op één plek in de database hoeft aan te passen. Zo nee, los dit dan op.

Uitwerking:

- a) -
- b) eigenaar, fietsenverhuurbedrijf, fiets, identificatienummer, framenummer, stadsfietsen, racefietsen, mountainbikes, e-bike, herenmodel, damesmodel, wielmaten, leverancier, bedrijfsnaam, woonplaats, adres, telefoonnummer.
- c) Identificatienummer FIETS, framenummer FIETS, herenmodel FIETS, wielmaat FIETS, bedrijfsnaam LEVERANCIER, woonplaats LEVERANCIER enzovoort.
- d) Entiteiten fiets, leverancier
- e) Entiteit fiets Attributen: id, framenummer, soort, uitvoering, model, wielmaat.
Entiteit leverancier Attributen: bedrijfsnaam, woonplaats, adres, telefoonnummer
- f) Entiteit fiets: uid fiets_id reserve uid framenummer
Entiteit leverancier: uid lev_id, reserve uid naam Naam is minder geschikt als uid

want hoewel je bedrijfsnaam bij de kamer van koophandel uniek moet zijn kan het zijn dat de leverancier dezelfde bedrijfsnaam heeft als een bedrijf dat langer dan een jaar is uitgeschreven bij de kamer van koophandel. Verder zou een leverancier zijn naam kunnen veranderen. Om die reden is ook telefoonnummer niet geschikt want telefoonnummers worden soms opnieuw uitgegeven. Ook het kamer van koophandel nummer is geen geschikt nummer voor uid omdat bij de verkoop van een bedrijf het bedrijf een ander kamer van koop handel nummer krijgt. Ook weet je maar nooit of de kamer van koophandel ieder bedrijf in de toekomst een ander nummer gaat geven. Je wilt niet afhankelijk van derden zijn. De kans dat de naam van de leverancier wordt aangepast is echter niet zo groot. Als je een tabel toestaat waarbij de oude en de nieuwe naam van een bedrijf voorkomt kan "naam" een reserve uid zijn. Een voordeel van "naam" als secondary UID is verder dat er een duidelijk verband is tussen deze UID en de naam van de tabel.

g)



h) Als je alle eigenschappen in één entiteit fiets had gestopt, en het adres van het bedrijf zou veranderen dan zou dit mogelijk op heel veel regels van de tabel moeten worden aangepast. Dit komt omdat de onderste vijf eigenschappen in het plaatje hieronder onderling afhankelijk zijn. Oplossing: aparte entiteit leverancier.



Practicum stap 2: Relaties, Erdish en ERD

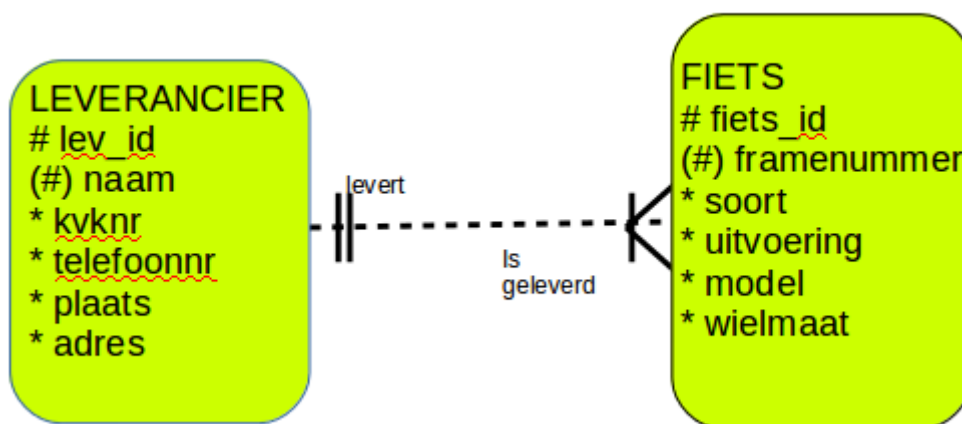
- Schrijf voor iedere relatie een zin op voor de kraaienpoot notatie.
- Teken de relaties in het ERD en schrijf de werkwoorden bij de relaties.

Uitwerking:

- Iedere leverancier levert één of meer fietsen.

Iedere fiets is geleverd door precies één leverancier.

- Het is een stippellijn omdat het kind (fiets) van de relatie niet de UID van de ouder overneemt.



Practicum stap 3: Normaalvormen en matrix

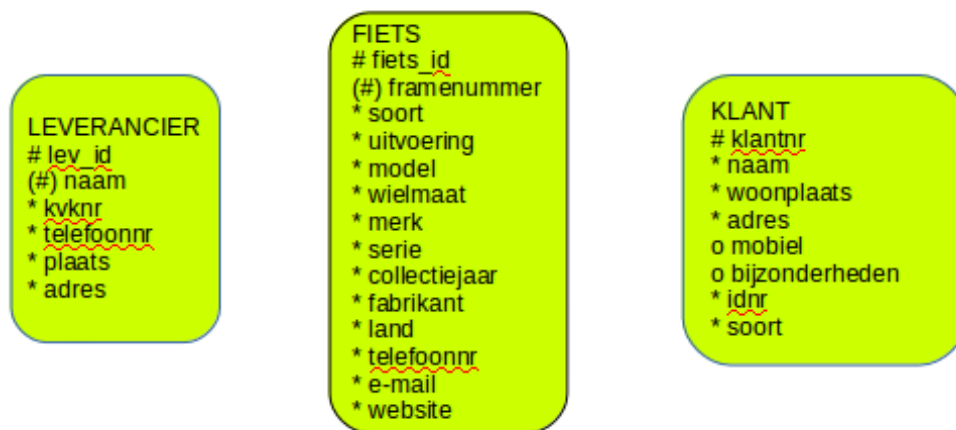
De eigenaar van de fietsenverhuur heeft nog meer wensen. Hij wil ook kunnen bijhouden welke fiets door welke klant gehuurd is. Van zijn klanten vraagt hij naam, woonplaats, adres, mobiel telefoonnummer en nummer van een identiteitsbewijs. Dat kan de identiteitskaart, paspoort of rijbewijs zijn. Soms noteert hij ook bijzonderheden. Een klant kan op zijn naam ook meerdere fietsen huren.

Verder noteert hij van iedere fiets de fabrikant, merk, serie en collectiejaar. Een fabrikant kan meerdere merken produceren. Hij noteert ook het land, telefoonnummer, e-mail en de website van de fabrikant. De leveranciers kopen hun fietsen bij de fabrikanten in.

- a) Markeer de belangrijke zelfstandige naamwoorden.
- b) Welke nieuwe entiteiten zijn nodig en geef ze een naam.
- c) Noteer bij iedere entiteit de attributen die nodig zijn.
- d) Is iedere voorbeeld van een entiteit uniek? Noteer bij iedere entiteit welke eigenschap als UID (unique identifier) kan fungeren. Noteer indien beschikbaar ook een secondary UID.
- e) Teken de entiteiten in een ERD (Entity Relationship Diagram) compleet met attributen en de tekens ervoor of ze de primary UID #, secondary UID (#), mandatory * (verplicht) of optional o (niet verplicht) zijn. Je hoeft nog geen verbindinglijnen tussen de entiteiten te trekken.
- f) Controleer of er afhankelijkheden tussen de attributen zijn. Zo ja, los dit dan op.
- g) Maak een matrix van de entiteiten die je nu hebt en noteer daar werkwoorden in voor de relaties tussen die entiteiten. Zet een kruisje als er geen relatie is.
- h) Ga met behulp van de tabel na of er misschien overbodige relaties in staan.
- i) Schrijf alle relaties op in zinnen voor de kraaienpootnotatie.
- j) Teken de relaties en schrijf de werkwoorden er bij.

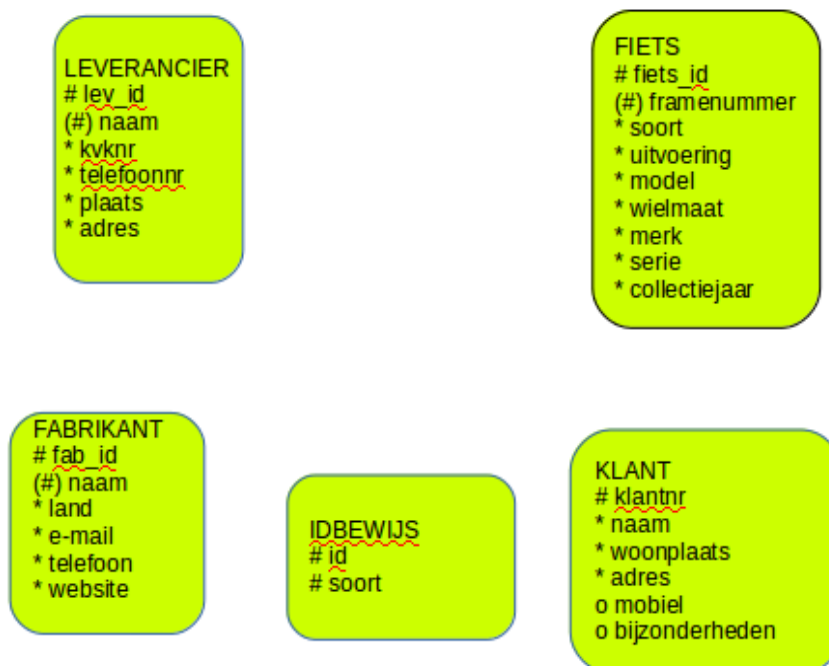
Uitwerking

- a) Klant, woonplaats, naam, adres, mobiel telefoonnummer, identiteitsbewijs, paspoort, identiteitskaart, rijbewijs, bijzonderheden, fabrikant, merk, serie, collectiejaar, land, fabrikant
- b) Entiteiten: klant (en fabrikant maar een aantal leerlingen zullen dit in eerste instantie niet hebben).
- c) Entiteit klant attributen: klantnr, naam, woonplaats, adres, mobiel, idnr, soort, bijzonderheden.
Entiteit fiets nieuwe attributen merk, serie, collectiejaar, fabrikant, land, telefoonnr, e-mail, website (we gaan er hier nog vanuit dat ze nog niet op het idee van een aparte entiteit fabrikant zijn gekomen).
- d) klant primary UID klantnr De eigenschap idnr is niet geschikt omdat een klant meerdere idnummers kan hebben van verschillende idbewijzen zoals rijbewijs en paspoort.
- e)



- f) Bij de entiteit FIETS zijn de onderste vijf attributen onderling afhankelijk. Oplossing: nieuwe entiteit FABRIKANT. Naam zou hier de secondary UID kunnen zijn om een duidelijk verband te hebben met de tabelnaam.

Ook bij klant doet zich een probleem voor. Zoals het er nu in staat kan dezelfde klant drie keer in het systeem voorkomen, iedere keer met een ander id_bewijs. Oplossing: nieuwe entiteit IDBEWIJS. Door de primary UID van deze entiteit een combinatie van combinatie van id en soort te laten zijn voorkomt dat als hetzelfde nummer van bijvoorbeeld een rijbewijs ook voorkomt op het paspoort van iemand anders, problemen kan geven.



g)

	fiets	leverancier	klant	fabrikant	idbewijs
fiets	x	Is geleverd	Is verhuurd	Is gemaakt	x
leverancier	levert	x	x	Koopt in	x
klant	huurt	x	x	x	heeft

fabrikant	maakt	verkoopt	x	x	x
idbewijs	x	x	Behoort tot	x	x

h) Je zou kunnen denken dat de relatie tussen fabrikant en fiets overbodig is omdat er al relaties zijn tussen fiets en leverancier en tussen leverancier en fabrikant.

Aangezien er tussen leverancier en fabrikant een meer meer relatie bestaat, is dan echter onmogelijk te achterhalen van welke fabrikant een fiets afkomstig is. n .

i) Iedere leverancier levert één of meer fietsen.

Iedere fiets is geleverd door precies één leverancier.

Iedere klant huurt nul of meer fietsen.

Iedere fiets is verhuurd aan nul of één klant.

Iedere leverancier koopt in bij één of meer fabrikanten.

Iedere fabrikant verkoopt aan één of meer leveranciers.

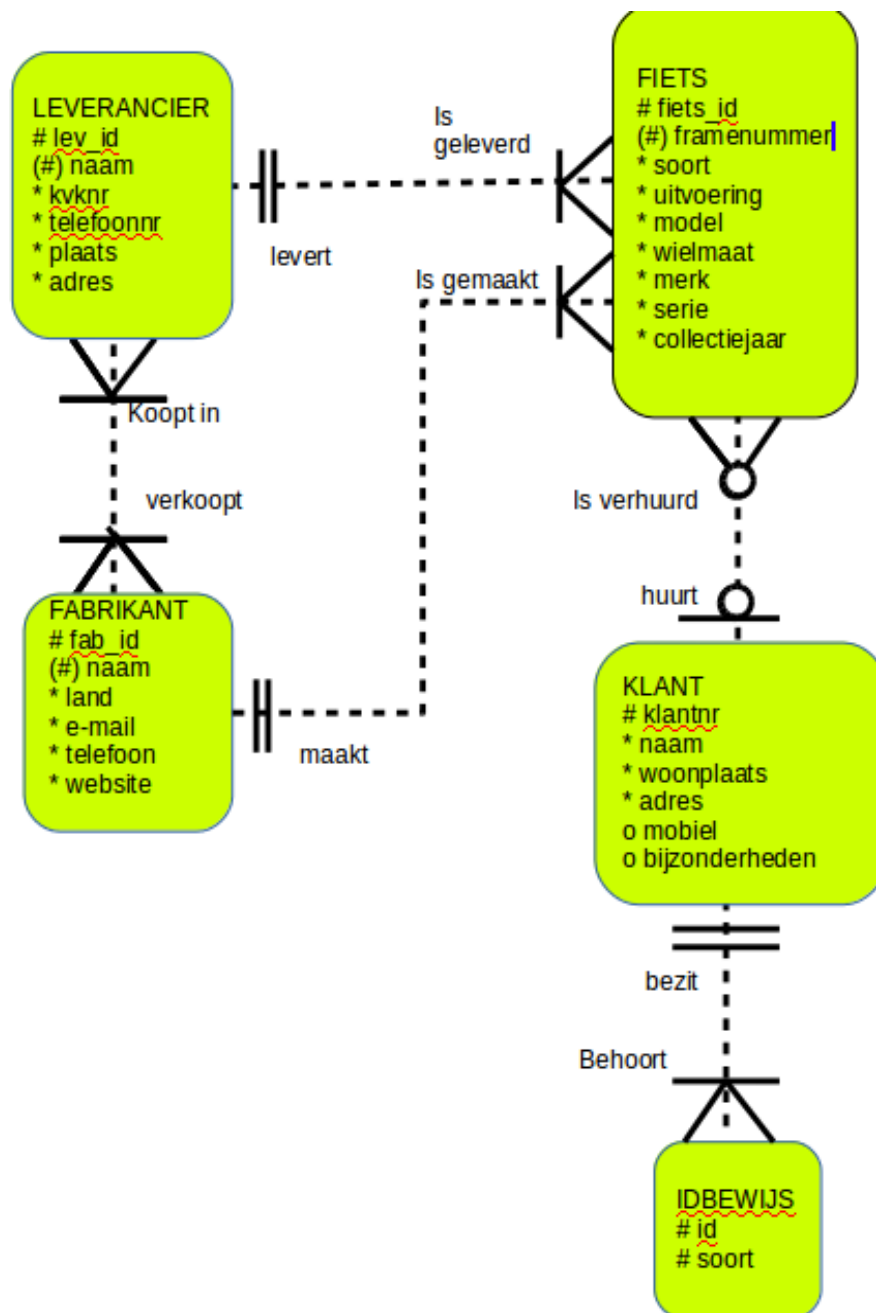
Iedere klant bezit één of meer idbewijzen.

Ieder idbewijs behoort tot precies één klant.

Iedere fiets is gemaakt door precies één fabrikant.

Iedere fabrikant heeft één of meer fietsen gemaakt.

j)



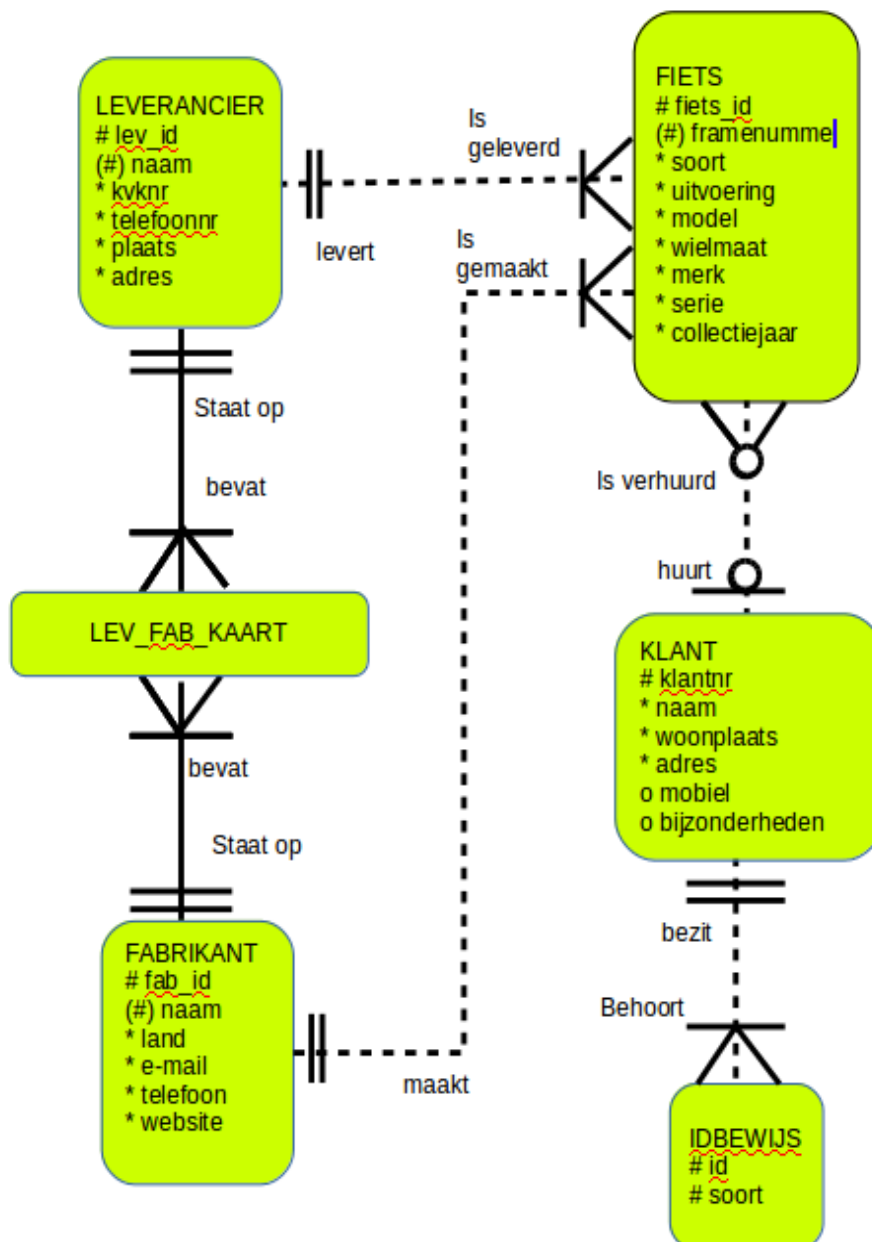
Practicum stap 4: Meer meer relaties oplossen

- In de ERD van stap 3 bevindt zich een meer meer relatie. Hier kunnen databases niet mee omgaan. Los dit probleem op en vermaak de ERD.
- Schrijf ook de kraaienpoot notatie zinnen opnieuw op.

Uitwerking

De oplossing is een intersection entiteit tussen LEVERANCIER en FABRIKANT. Je zou deze intersection entiteit bijvoorbeeld LEV_FAB_KAART kunnen noemen. Dit is een kaartje waarop precies één leverancier en één fabrikant staat. De lijnen zijn doorgetrokken omdat LEV_FAB_KAART zijn uid ophaalt uit LEVERANCIER en uit FABRIKANT. De uid van de intersection entiteit is dus leverancier_id + fabrikant_id.

a)



b) Iedere leverancier levert één of meer fietsen.

Iedere fiets is geleverd door precies één leverancier.

Iedere klant huurt nul of meer fietsen.

Iedere fiets is verhuurd aan nul of één klant.

Iedere leverancier staat op één of meer lev_fab_kaarten.

Iedere lev_fab_kaart bevat precies één leverancier.

Iedere fabrikant staat op één of meer lev_fab_kaarten.

Iedere lev_fab_kaart bevat precies één fabrikant.

Iedere klant bezit één of meer idbewijzen.

Ieder idbewijs behoort tot precies één klant.

Iedere fiets is gemaakt door precies één fabrikant.

Iedere fabrikant heeft één of meer fietsen gemaakt.

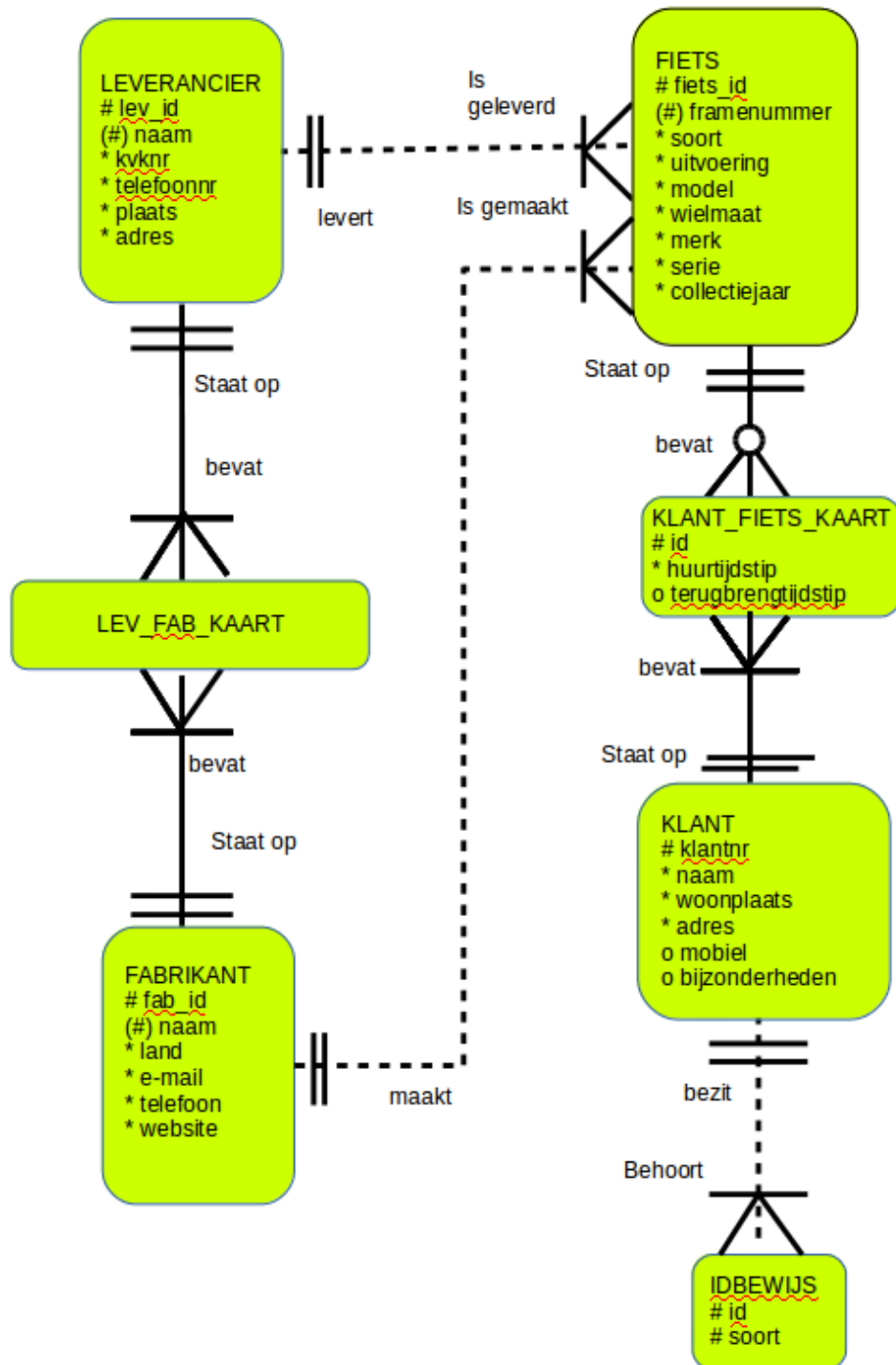
Practicum stap 5: Geschiedenis

De eigenaar van de fietsenverhuur zou graag kunnen zien welke klanten in het verleden een bepaalde fiets hebben gehuurd.

- a) Pas de ERD aan zodat dit mogelijk wordt.
- b) Schrijf ook de kraaienpoot notatie zinnen opnieuw op. Verstandig is om dit te doen voordat je de nieuwe relaties gaat tekenen.
- c) Welke van de twaalf relaties die je dan hebt zijn niet overdraagbaar?

Uitwerking

- a) Hiervoor maken we een intersection entiteit die we bijvoorbeeld KLANT_FIETS_KAART noemen. Deze haalt zijn UID deels op uit FIETS en KLANT (doorgetrokken lijnen). Daarnaast behoort huurtijdstip tot de UID. Terugbrengtijdstip is optioneel omdat dit pas ingevuld kan worden als de fiets is teruggebracht. Een fiets kan nog geen geschiedenis hebben als hij nog niet is verhuurd. Je wordt klant als je voor het eerst een fiets huurt.
- b) Iedere leverancier levert één of meer fietsen.
Iedere fiets is geleverd door precies één leverancier.
Iedere klant staat op één of meer klant_fiets_kaarten.
Iedere klant_fiets_kaart bevat precies één klant. *
Iedere fiets staat op nul of meer klant_fiets_kaarten.
Iedere klant_fiets_kaart bevat precies één fiets. *
Iedere leverancier staat op één of meer lev_fab_kaarten.
Iedere lev_fab_kaart bevat precies één leverancier. *
Iedere fabrikant staat op één of meer lev_fab_kaarten.
Iedere lev_fab_kaart bevat precies één fabrikant. *
Iedere klant bezit één of meer idbewijzen.
Ieder idbewijs behoort tot precies één klant.
Iedere fiets is gemaakt door precies één fabrikant.
Iedere fabrikant heeft één of meer fietsen gemaakt.
- c) De relaties met een ster erachter zijn niet overdraagbaar. Dit kan niet in de ERD worden opgenomen maar moet wel in de ERM documentatie staan. In de praktijk betekent dit dat er voor de gebruikers geen vensters open mogen staan waarin de KLANT_FIETS_KAART en LEV_FAB_KAART aangepast kunnen worden.



Practicum stap 6: Supertype, subtype

De eigenaar van de fiets verhuur wil graag dat bij bakfietsen ook de breedte in het systeem worden opgenomen. Bakfietsen die breder zijn dan 75 cm mogen namelijk op de openbare weg, ook al is er een verplicht fietspad. Ook het aantal wielen van een bakfiets moet in het systeem. Er zijn namelijk tweewielers en driewielers. Bij tandems wil hij dat het aantal zitplaatsen en de lengte van de tandem in het systeem komt.

- a) Gebruik het principe van supertype, subtype om dit in je ERD te verwerken.

Uitwerking

