# Portfolio LU1 Ontwerp Relationele Datapersistentie

**Datum:** [datum]

**Groep:** 22INF1A-5

1. Algemeen

In dit deel van de proftaak werken we een database uit voor de promotionele website van ons fictieve bedrijf PinPointParking. In dit document wordt toelichting gegeven op het ERD en het relationeel model die te vinden zijn in dit zip-bestand.

Voor het opstellen en aanpassen van de .sql files hebben we gebruikgemaakt van de editor en GIT. Voor het verzamelen van alle files en updates hebben we Github gebruikt, de repository is te vinden op <https://github.com/Inge360/proftaak-1A5>

### 2. Inhoud van ZIP-bestand [naam]

- Bestand *Portfolio LU1 Ontwerp Relationele Datapersistentie 22INF1A-5.docx*

- Bestand *[ERD]*

- Bestand *[Relationeel Model]*

### 3. Aanpak

Allereerst hebben we in kaart gebracht welke data we wilden opnemen in de database. Welke data zijn echt nodig voor het uitvoeren van de dienst. Op basis daarvan hebben we de onderdelen verdeeld om verder uit te werken en vervolgens elkaars werk bekeken, getest, aangevuld en aangepast.

4. Acceptatiecriteria

Definition of Done Datapersistentie

Dit onderdeel van PinPointParking is af als:

* het ERD de onderneming volledig en juist in kaart gebracht zijn.
* de ERD goed omgezet is naar het Relationeel Model.
* het Relationeel Model goed is omgezet naar de Data Definition Language.
* de referentiële integriteit goed is getest en werkt volgens de eis dat persoonsgegevens van de klant en bedrijf verwijderd kan worden zonder de gegevens in de tabellen voor de facturering te verliezen.
* met DCL users db\_admin, Medewerker, Bedrijf en Particulier aangemaakt zijn.
* met DCL wordt gecombineerd met Views die databasebeperkingen instellen voor de user Bedrijf en Particulier zodat zij geen data kunnen zien/bewerken van anderen.
* met DCL rollen gemaakt zijn voor meerdere users binnen de groep db\_admin en Medewerker.
* met DML de vooraf user stories uitgewerkt zijn.
* met DML enkele Stored Procedures gemaakt zijn waarmee complexere berekeningen van de parkeerkosten gemaakt kunnen worden.
* met DML Events gemaakt zijn die blabla
* met DML Triggers gemaakt zijn die blabla

Users

1. *db\_admin*

Oprichters en developers van PinPointParking die toegang hebben tot alle data en alle bewerkingen kunnen doen.

1. *Medewerker*

Medewerkers van PinPointParking die administratieve taken hebben en alleen data kunnen zien en verwijderen vanwege de AVG.

1. *Bedrijf*

Zakelijke klant die medewerkers gebruik laat maken van PinPointParking, alle facturen gaan naar de financiële afdeling van het bedrijf. Zij kunnen alleen bij de data die samenhangt met hun eigen medewerkers. Deze kunnen ze zien, maken, aanpassen, verwijderen.

1. *Gebruiker*

Particuliere klanten die zelf hun facturen betalen. Zij kunnen alleen bij de data die samenhangt met hun eigen account. Deze kunnen zij zien, maken, aanpassen, verwijderen.

User Stories medewerker

* Als medewerker van PPP wil ik de exacte parkeerkosten weten van een reservering zodat ik de factuur kan berekenen.
* Als medewerker van PPP wil ik alle facturen ophalen met naam en adres zodat ik zie wie facturen moet krijgen.
* Als een medewerker van PPP wil ik alle parkeeromgevingen opsommen van veel naar weinig gebruikt zodat ik weet waar mogelijk behoefte is aan extra parkeergelegenheid.
* Als een medewerker van PPP wil ik de klant ophalen die het meest heeft geparkeerd zodat we daarmee rekening kunnen houden met kortingsacties.
* Als een medewerker van PPP wil ik de bedrijven zien met het aantal medewerkers van veel naar weinig zodat we daarmee rekening kunnen houden met abonnementsaanbiedingen.
* Als een medewerker van PPP wil ik een beknopt overzicht van klant, kenteken en het aantal sessies per klant zodat we snel zien hoe vaak klanten PPP gebruiken.

User Stories db\_admin

* Als admin wil ik de processlist zien van alle users zodat ... - ??

User Stories bedrijf

* Als bedrijfsmedewerker wil ik de factuur van een medewerker ophalen inclusief de parkeerkosten zodat we specifieke kosten in kaart kunnen brengen.
* Als bedrijfsmedewerker wil ik een overzicht met alle NAW-gegevens van alle medewerkers die PPP gebruiken zodat we overzicht houden over de administratie.

User Stories particulier

* Als particulier wil ik al mijn parkeersessies zien zodat ik weet hoeveel facturen ik nog zal ontvangen – nog doen.
* Als particulier wil ik mijn adres wijzigen zodat de gegevens kloppen na mijn verhuizing.
* Als particulier wil ik mijn account verwijderen zodat ik controle heb over mijn persoonsgegevens – nog doen.

### 5. ERD en beschrijving

Zie ERD, bestandsnaam *[…]*

Toelichting soort relaties (1:1, n:m etc) en intersectiedata.

### 6. Relationeel model en documentatie

Zie Relationeel model, bestandsnaam *Relationeel Model.png*

Toelichting normaalvormen

**Tabellen: Klant, Zakelijk, Adres, Voertuig, ParkeerSessie, ParkeerPlaats, Factuur en FactuurRegel**

Klant

*KlantId* – deze integer word gekoppeld aan een persoon en alle data kan via deze Id worden opgeroepen.

*Registratiedatum* – Wanneer er onverhoopt een datalek is, kunnen we alle gebruikers van voor die specifieke datum op de hoogte stellen. Daarnaast kunnen we hiermee een telling bijhouden hoeveel nieuwe accounts er zijn per periode.

*Mailadres* – Dit is nodig om de klant up to date te houden met facturen en mogelijk nieuws. *Wachtwoord* – Dit is een varchar128 waar in een SHA encryptie zal worden opgeslagen.

*Voornaam, tussenvoegsel en Achternaam* – De naam van de klant.

*Mobiel* – Een extra contactmogelijkheid, mocht het mailadres niet meer werken of als de klant niet reageert op mails enzovoort. Daarnaast kunnen klanten gebruik maken van notificaties vanuit de app.

*Zakelijk* – Deze boolean geeft aan of de klant een bedrijf is of een particulier.

*KvkNummer* – Deze is gekoppeld aan de tabel Zakelijk, als een klant niet zakelijk is blijft deze null. *MachtigingAutoInc* – Dit is om aan te geven of de klant gekozen en toestemming gegeven heeft voor een automatische incasso.

*MachtigingDatum* – De datum waarop de machtiging is afgegeven.

*IBAN* – Het bankrekeningnummer van de klant.

Zakelijk

*KvkNummer* – Uniek bedrijfsnummer wat gekoppeld is aan de Klant tabel.

*BedrijfsNaam* – Naam van het bedrijf, om de factuur persoonlijker te maken en bij troubleshooting een erg handige extra.

*BtwNummer* – Een uniek niet verplicht nummer, eventueel om het bedrijf te verifiëren. Omdat het niet verplicht is ‘eventueel’ toegevoegd.

Adres

*AdresId* – Integers communiceren sneller met elkaar waardoor we hier hebben gekozen voor een int in plaats van een unieke varchar of composite key. Mochten we ook met meerdere adressen willen gaan werken dan gaat dat ook gemakkelijker nu.

De rest van de attributen zijn vanzelfsprekend, dit zijn adresgegevens voor de factuur.

Voertuig

*VoertuigId* – Auto’s kunnen gedeeld worden, daarom hebben we een Id aangemaakt in plaats van het kenteken te gebruiken als primary key. Ook is een integer die met integer gelinkt is sneller.

*Kenteken* – Deze is nodig voor de factuur zodat de klant weet voor welk voertuig de factuur is.

*KlantId* – De koppeling tussen klant en voertuig, dit is ook een 1-op-meer relatie omdat een klant ook meerdere voertuigen kan hebben. Moet dit dan niet een meer-op-meer relatie zijn? Omdat auto’s gedeeld kunnen worden en een klant meerdere voertuigen kan hebben?

ParkeerSessie

*ReserveringsNr* – Een voertuig kan maar één Parkeersessie tegelijk hebben (dit wordt tegen gehouden in de code) maar wel meerdere Parkeersessies in een dag/week enzovoort. *BeginTijd* – De begintijd van de parkeersessie om te berekenen wat de kosten zijn *EindTijd* – Vanzelfsprekend om de kosten te berekenen

ParkeerPlaats

*ParkeerPlaatsId* – Voor de snelle link tussen de tabellen en een uniek nummer.

*Straat, Postcode, Stad* – Deze attributen worden gebruikt om de parkeerplaats mee te identificeren.

(Als dat is wat je bedoelt. Ik snapte jouw versie nog niet helemaal: “een soort parkeeromgeving te hebben waar de kosten worden berekend voor de omgeving.” De berekening zit bij tariefuur en begin-/eindbetaaldparkerendag)

*HuisNR* – Deze is niet verplicht omdat parkeren in een wijk geen huisnummer heeft. Sommige grote garages of parkeerterreinen hebben ook geen huisnummer.

*ParkeerTerrein* – Een boolean of het gaat om een terrein of niet.

*ParkeerGarrage* – Een boolean of het gaat om een garage of niet.

*TariefUur* – De parkeerkosten per uur van de betreffende parkeerplaats.

*MaxDuur* – Op sommige parkeerplaatsen geld een parkeerlimiet, met deze optionele optie wordt dat dan aangegeven.

*BeginBetaaldParkerenDag* – De begintijd van het betaald parkeren. Deze is nodig om te berekenen wanneer de betaling in zal gaan of nodig zal zijn.

*EindBetaaldParkerenDag* – De eindtijd van het betaald parkeren.

Factuur

*KlantId* – De klant kan meerdere facturen hebben daarom zit er een link tussen de tabel Klant en de Factuur tabel. In deze hele tabel worden de gegevens plat opgeslagen, ze worden op het moment opgehaald dat de factuur wordt gemaakt, bijvoorbeeld aan het einde van elke maand. Op dat moment worden alle gegevens die op dat moment bekend zijn in de database opgeslagen en krijgt de tupel een factuurnummer. Deze is dan weer gekoppeld aan het KlantId.

*FactuurNr* – Dit nummer is wettelijk verplicht en is ook een mooi nummer om als primary key te gebruiken. De rest van de gegevens worden opgehaald uit de andere tabellen: Klant, Zakelijk, Adres.

FactuurRegels

*FactuurNr* – Deze heeft een meer-op-1 relatie met Factuur omdat via FactuurRegel meerdere sessies worden opgehaald dus meerdere sessies worden vervolgens weergeven in de factuur waar een totaalprijs wordt opgemaakt van alle sessies (factuurregels). Bijvoorbeeld: Factuur 10001 heeft 5 parkeersessies met 5 verschillende reserveringsnummers.

*ReserveringsNr* – Deze wordt opgehaald van de tabel ParkeerSessie>ReserveringsNr, zit geen link in de database omdat deze gegevens veranderen en je voor de factuur een soort snapshot moet hebben.

*Kenteken* – Deze wordt opgehaald bij de ParkeerSessie via het ReserveringsNR>VoertuigId en is nodig voor de factuur. Per sessie is er een kenteken, dit is zodat iemand meerdere auto’s in een maand op zijn factuur kan hebben.

*BeginTijd, EindTijd, TariefUur, MaxDuur, BeginBetaaldParkerenDag en EindBetaaldParkerenDag –* Deze worden gebruikt om te berekenen wat de prijs zal worden voor de parkeersessie, dit wordt dan wel gedaan via SQL aggregatie maar deze gegevens zouden genoeg moeten zijn. Ook als prijzen in de toekomst veranderen, zal in de factuurregels alsnog de prijzen van dat moment bewaard blijven zodat het klopt.

*Straat, HuisNr, Stad, Postcode* – Dit zijn de gegevens van waar er geparkeerd is in de sessie.