

Verslag IT Databank maken

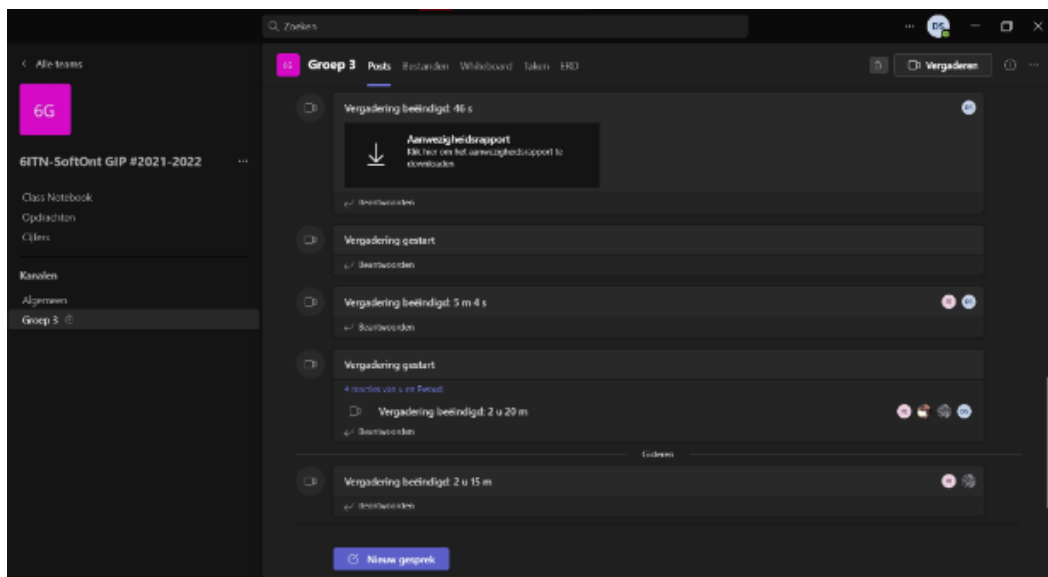
Inhoudstabel

1. SAMENWERKING:	2
1.1. TEAM AANMAKEN	2
2. HET BEGIN	2
2.1. IDEE OPBOUWEN	2
2.2. ENTITEITEN	3
3. ERD-MODEL	3
3.1. LUCIDCHART	3
3.2. ERD-MODEL AFWERKING	4
4. NORMALISEREN	5
4.1. NORMALISATIEPROCES	5
5. MYSQL WORKBENCH	6
5.1. INTEGRITEITSREGELS	6
5.2. MAKEN VAN DATABANK	6
6. ONZE ERVARING	8
6.1. ONZE ALGEMENE ERVARING	8
6.2. MOEILIKHEDEN	8
7. QUERY'S	9
8. KLADNOTITIES (INZICHT IN ONS PROJECT...)	14
8.1. BEGIN VERSLAG	14
8.2. ALGEMEEN	15

1. Samenwerking:

1.1. Team aanmaken

We voelde aan dat de samenwerking niet zo efficiënt verliep via een groep chat in teams. Onze oplossing hiervoor was een eigen team maken in Teams, het probleem was dat onze teams accounts niet de nodige rechten hadden om een eigen kanaal te creëren. We namen hiervoor initiatief en vroegen dit aan De vakleerkracht. We konden elkaar nu thuis verbeteren of samenwerken aan meerdere taken. Het werd voor ons dan ook gemakkelijker om vergaderingen te plannen, waardoor we efficiënter konden samenwerken.



2. Het Begin

2.1. Idee opbouwen

Bij de start van ons idee begonnen we met een naam te verzinnen voor ons “bedrijf”, na wat discussiëren kwamen we op de naam: “Total Tickets”. Ons bedrijf is een ticketsysteem voor bedrijven zoals diegene waar wij stage liepen.

Onze klanten (bedrijven) hun klanten kunnen bij ons tickets aanmaken.

De klanten van de bedrijven die een contract hebben bij ons kunnen tickets bij ons aanmaken

Het kan zijn dat onze klanten een gemeenschappelijke klant hebben

De prioriteit van de klant en mogelijk om in bepaalde uren en op bepaalde dagen een ticket aan te maken wordt door het ContractLevel bepaald.

met hun problemen, deze zijn ingedeeld in verschillende levels (Tiers).

In de foto hiernaast, kan je duidelijk zien hoe wij dit idee snel hebben geschetst.

2.2. Entiteiten

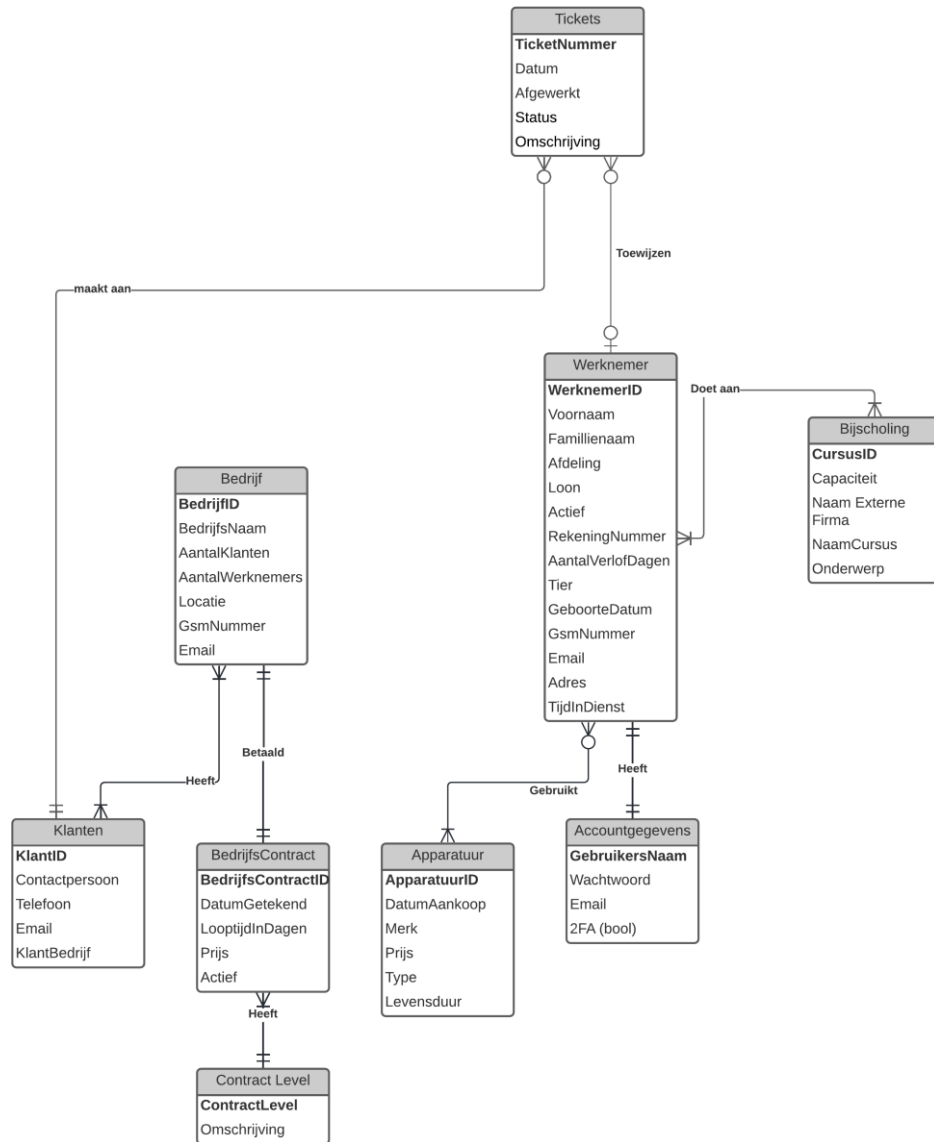
De volgende stap die we deden waren entiteiten op papier invullen zodat we de basis entiteiten voor later al hebben. Een entiteit is een algemene weergave van één object in een tabel, die we later aanvullen met gegevens (attributen).

3. ERD-model

3.1. LucidChart

We hebben het ERD-model geïntegreerd in LucidChart: dit is een webapplicatie waarmee je gemakkelijk diagrammen kunt maken voor zulke toepassingen. We kunnen LucidChart makkelijk integreren in Teams en dit bevordert de samenwerking, dit is de redenering achter onze keuze van dit programma. Nadat we onze entiteiten hebben ingevuld hebben we ook nog een deel van de relaties aangemaakt.

[ERD-Model online versie](#)



3.2. ERD-Model afwerking

De volgende stap voor ons ERD-Model is de attributen bij de passende entiteiten toevoegen. We probeerde zo veel mogelijk attributen in te vullen zodat we een realistisch databank kunnen simuleren. We kiezen wel enkel realistische gegevens die in een echt bedrijf noodzakelijk zouden zijn. Maar sommige zijn als extra toegevoegd: bv. De gebruikersnaam hebben we gekozen zodat collega's makkelijk kunnen communiceren met elkaar. De rest spreekt allemaal voor zich.

4. Normaliseren

4.1. Normalisatieproces

Om onze databank zo efficiënt mogelijk te maken moeten op ons ERD-model eerst het Normalisatieproces toepassen. Wat houdt dit nu juist in?: In dit proces zijn 3 stappen. Deze stappen bezorgen je wat meer werk, maar vanaf dat de databank in gebruikstelling wordt gezet zal deze minder werk vereisen. We zorgen er met dit proces voor dat alles goed aan elkaar gelinkt is en dat er geen gegevens dagelijks aangepast moeten worden. (Bv. Aantal tickets zou een gegeven kunnen zijn, maar dit zou elke dag aangepast moeten worden om actueel te blijven. Dit bezorgt ons meer werk en zou verdwijnen in dit proces.)

Door onze uitgebreide databank en [ERD-model](#), hadden we al voor ogen dat ons normalisatieproces ook groot ging zijn en zoals u kunt zien was dat ook zo (zie screenshot hiernaast). Het duurde ook 6-8 uur om dit klaar te krijgen.

5. MySQL Workbench

5.1. Integriteitsregels

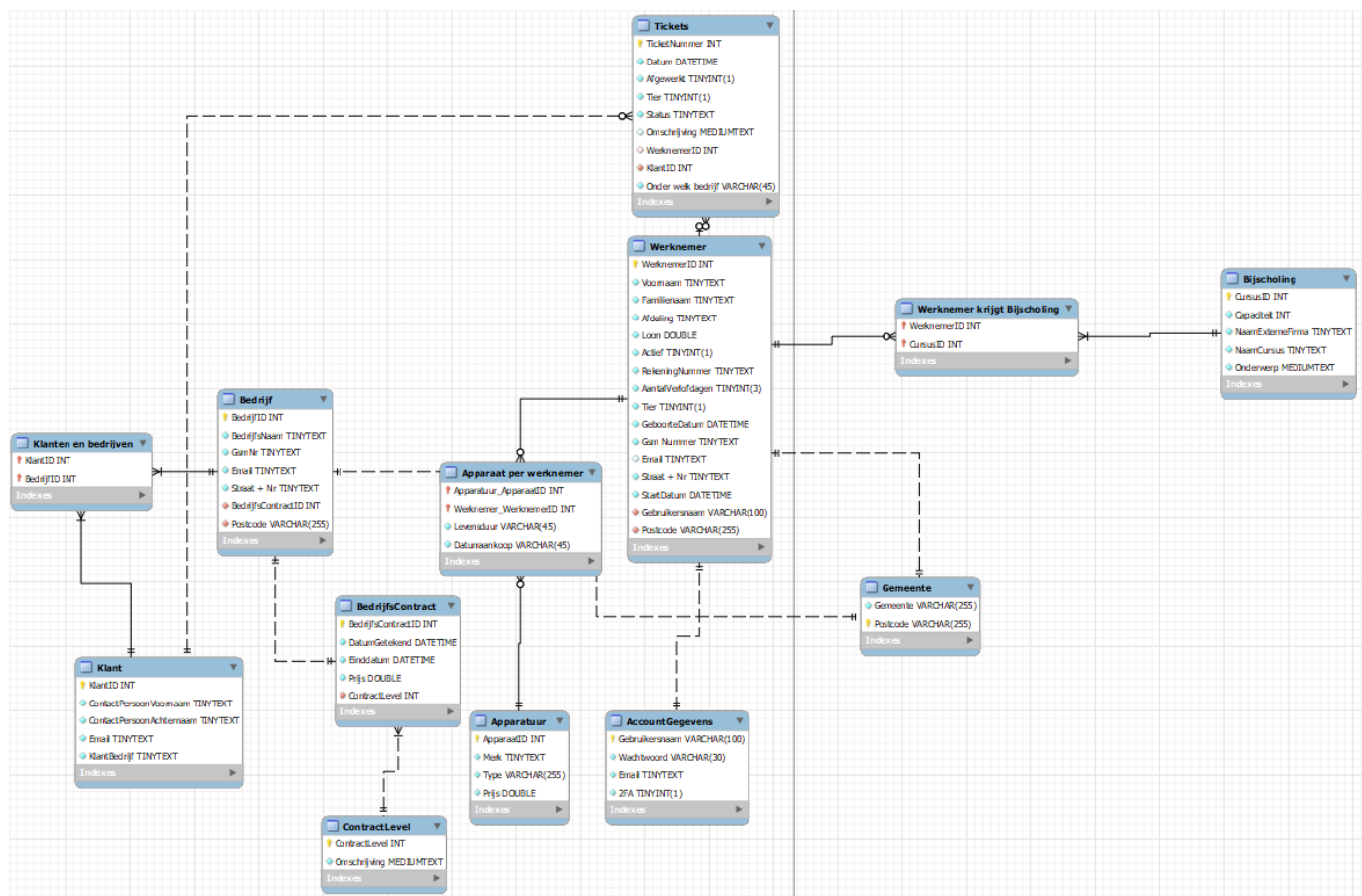
Als er iets wordt aangepast of verwijderd in de databank dat gelinkt is aan een ander gegeven op een andere plaats in dezelfde databank (refererende sleutel). Dan zorgen de integriteitsregels ervoor dat dit op een goede en geplande manier gebeurt, zodat er geen problemen gecreëerd worden met het verliezen of veranderen van gegevens. Je kan tussen een aantal acties kiezen als een van deze handelingen worden uitgevoerd.:

Bij aanpassen van gegevens	Bij verwijderen van gegevens
Cascade: Updaten van gegevens	Cascade: Verwijderen van beide gegevens
No Action: De gegevens in de gelinkte tabel blijven hetzelfde	No Action: De gegevens in de gelinkte tabel blijven hetzelfde
Restricted: niet toestaan (zo moet je dus zeker dubbel checken en kunnen er geen fouten gebeuren.)	Restricted: niet toestaan (zo moet je dus zeker dubbel checken en kunnen er geen fouten gebeuren.)
Set null: De gegevens in de gelinkte tabel leeg maken	Set null: De gegevens in de gelinkte tabel leeg maken

Voor onze integriteitsregels hebben wij gekozen om bij aanpassingen bij alles voor cascade te gaan, zo blijft onze databank heel efficiënt. En bij verwijderen hebben we vooral restricted want in ons geval zouden er enkel in uitzonderlijke gevallen gegevens verwijderd moeten worden, zo is het verplicht dat als je iets wil verwijderen je altijd moet nakijken dat het zeker juist is. Zo kunnen er geen ongelukken of fouten gebeuren.

5.2. Maken van databank

Om te beginnen hebben we de entiteiten vanuit ons [Normalisatieproces](#) toegevoegd aan het Databank ontwerp, vervolgens hebben we de attributen in deze tabellen bijgevoegd. Dan hebben we de relaties gelegd, hierbij moesten we rekening houden met de optionaliteit en cardinaliteit vanuit ons ERD-Model. Vervolgens hebben we de [Integriteitsregels](#) toegepast. Databank-ontwerp: (je kan best het bestand .mwb openen want de foto kan niet groter gezet worden...)



Daarna hebben we met behulp van Forward Engineering dit Databankontwerp omgezet naar een volwaardige Databank.

Tenslotte hebben we gegevens moeten verzinnen om deze daarna in de Databank te kunnen implementeren om hier [Query's](#) op uit te voeren.

6. Onze ervaring

6.1. Onze algemene ervaring

De opdracht in het algemeen is goed verlopen, we hebben hier dan ook met plezier aan gewerkt en hebben dit ook veel en consequent gedaan. We waren over het algemeen heel tevreden van deze opdracht en ook van ons resultaat bij de eerste opdracht.

Normaliseren was moeilijker dan gedacht. We hadden geen tabel om als voorbeeld te gebruiken, omdat we nog geen gegevens hadden in onze Databank. Het was lastiger om de normalisatie te voorstellen met bepaalde waardes. Tot nu toe hebben we altijd gewerkt met een ingevulde databank om te vertrekken, deze moesten we dan altijd verbeteren. Het was eens een verfrissende verandering om vanaf niets te beginnen.

6.2. Moeilijkheden

We hebben ondervonden dat je geen Primary Key die gelinkt is aan een andere tabel via een Foreign Key het gegevenstype TinyText mag geven. De reden hiervoor is dat MySQL Workbench niet kan inschatten hoe groot de waarde in deze Primary Key zou kunnen zijn. Daarom gebruiken we VarChar() met als limiet een heel hoog getal.

Dit komen we normaal niet tegen, omdat we altijd ID's als Primary Key gebruiken en dit is een getal, maar nu gebruiken we tekst als Primary Key. Tijdens het invullen van de Databank kregen we foutmeldingen, dit komt doordat de tabellen in een bepaalde volgorde ingevuld moet worden, bepaalde attributen hangen af van andere attributen in andere tabellen.

7. Query's

Hier kan je al onze query's uitgevoerd zien worden, elke query is meermaals getest en werkt dus volledig

Query:

```
SELECT WerknemerID, Voornaam, Familiennaam, Afdeling, Tier, Gebruikersnaam  
FROM werknemer  
WHERE Actief IS TRUE;
```

Reden van query: Alle werknemers die op dit moment actief zijn weergeven, zo heb je een goed beeld op wie je op een taak kan zetten en wie niet. Of wie je als collega om hulp kan vragen via de gebruikersnaam. Je kan deze query ook omdraaien als je bijvoorbeeld iemand extra nodig hebt die op dit moment niet actief is.:

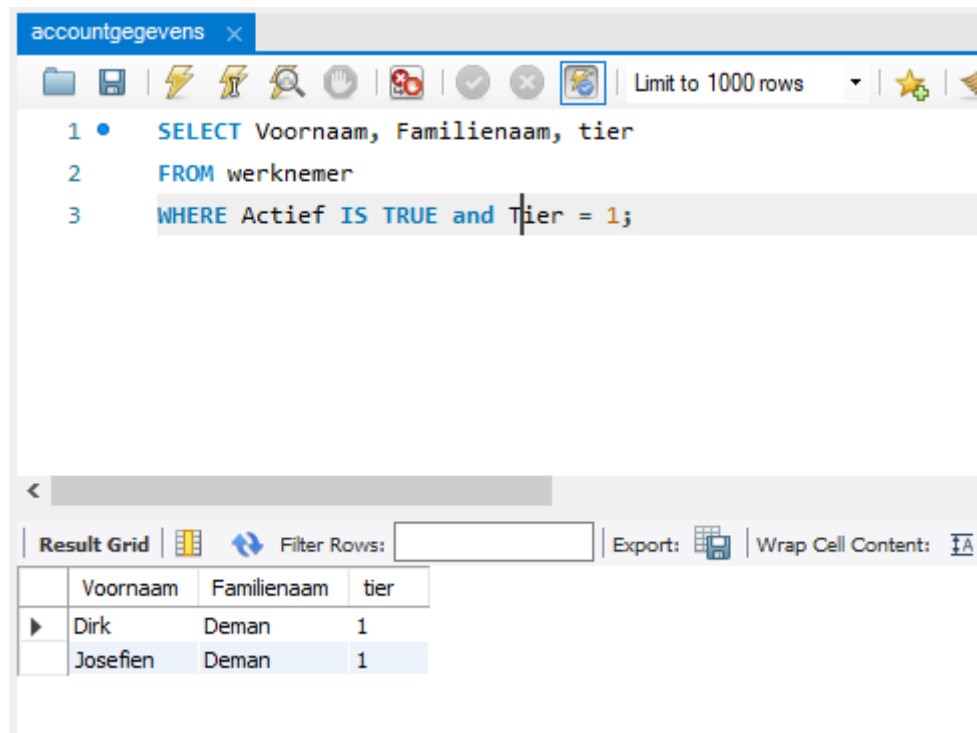
```
1 • SELECT WerknemerID, Voornaam, Familiennaam, Afdeling, Tier, Gebruikersnaam  
2 FROM werknemer  
3 WHERE Actief IS TRUE;
```

Result Grid						
Filter Rows:		Edit: Export/Import: Wrap Cell				
	WerknemerID	Voornaam	Familiennaam	Afdeling	Tier	Gebruikersnaam
1	1	Ewoud	Forster	Software	3	Ewoud_Totaltickets
2	2	Michiel	Van Den Bempt	Software	3	Michiel_Totaltickets
3	3	Sebastiaan	Smit	Netwerk	2	Sebastiaan_Totaltickets
10	10	Dirk	Deman	Netwerk	1	Dirk_Totaltickets
11	11	Josefien	Deman	Basishulp	1	Josefien_Totaltickets
	NULL	NULL	NULL	NULL	NULL	NULL

Query:

```
SELECT Voornaam, Familiennaam, tier  
FROM werknemer  
WHERE Actief IS TRUE and Tier = 1;
```

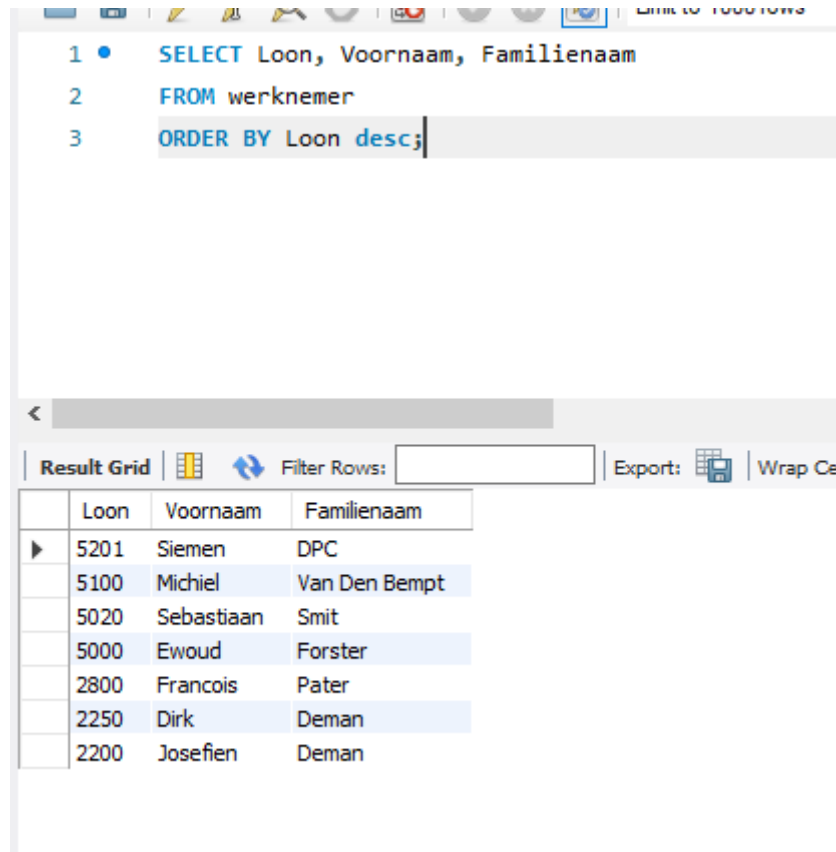
Reden van query: Specifiek kijken welke tier 1 werknemers er actief zijn. Handig voor de ICT Consultant om te checken ofdat er wel genoeg mensen in tier 1 zitten. (of dat er meer werknemers nodig zijn in deze tier) Ook cruciaal voor collega's van een andere tier om hun taak aan een andere collega van dezelfde of een andere tier door te geven



Query:

```
SELECT Loon, Voornaam, Familienaam
FROM werknemer
ORDER BY Loon desc;
```

Reden van query: De werknemers hun loon weergeven en hierop van hoog naar laag sorteren. Dit kan handig zijn om de lonen te vergelijken en om te zien of deze mensen het waard zijn.



```

1 • SELECT Loon, Voornaam, Familiennaam
2   FROM werknemer
3   ORDER BY Loon desc;

```

	Loon	Voornaam	Familiennaam
▶	5201	Siemen	DPC
	5100	Michiel	Van Den Bempt
	5020	Sebastiaan	Smit
	5000	Ewoud	Forster
	2800	Francois	Pater
	2250	Dirk	Deman
	2200	Josefien	Deman

Query:

```

SELECT Voornaam, AantalVerlofdagen,Afdeling
      FROM werknemer
      WHERE Actief IS TRUE
      ORDER BY AantalVerlofdagen desc;

```

Reden van query: Verlof dagen van werknemers zien, dit kan handig zijn om te zien hoelang je zonder iemand kan zitten, als dit een te lange periode is kan je beter iemand extra hiervoor in dienst nemen. Dan hoeft je nooit zonder iemand uit deze afdeling te zitten.

```

1  SELECT Voornaam, AantalVerlofdagen,Afdeling
2  FROM werknemer
3  WHERE Actief IS TRUE
4  ORDER BY AantalVerlofdagen desc;

```

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	Voornaam	AantalVerlofdagen	Afdeling
▶	Sebastiaan	26	Netwerk
	Ewoud	24	Software
	Michiel	23	Software
	Dirk	10	Netwerk
	Josefien	2	Basishulp

Query:

```

SELECT BedrijfsNaam, contractlevel, Omschrijving
FROM bedrijf INNER JOIN contractlevel ON BedrijfID = ContractLevel

```

Reden van query: Zien welke bedrijven welke Contractlevel heeft. Zo kan je zien als er een ticket binnen komt hoe snel je deze mag oplossen.

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	BedrijfsNaam	contractlevel	Omschrijving
▶	Get Nv	1	3 van de 5 Werkdagen van 9 tot 5
	Orange Cyberdefense	2	5 van de 5 Werkdagen van 9 tot 5
	Tcog	3	7 van de 7 Werkdagen van 9 tot 5
	VanRoey.be	4	5 van de 7 Werkdagen 24 uur

Query:

```
SELECT DISTINCT Tickets.TicketNummer, Tickets.Datum, Tickets.Tier, Tickets.Status,  
                Tickets.WerknemerID, Werknemer.Gebruikersnaam,  
                Werknemer.Voornaam, Werknemer.Familienaam  
FROM Tickets INNER JOIN Werknemer ON Werknemer.WerknemerID = Tickets.werknemerID,  
werkknemer Werknemer2 RIGHT JOIN AccountGegevens ON Werknemer2.Gebruikersnaam =  
                AccountGegevens.Gebruikersnaam  
ORDER BY Tickets.Datum DESC;
```

Reden van query: Een werknemer wilt de Gebruikersnaam en naam van de toegewezen
Collega's op het recentste ticket weten om deze te contacteren:

```
1  SELECT DISTINCT Tickets.TicketNummer, Tickets.Datum, Tickets.Tier, Tickets.Status,  
2  Tickets.WerknemerID, Werknemer.Gebruikersnaam,  
3  Werknemer.Voornaam, Werknemer.Familienaam  
4  FROM Tickets INNER JOIN Werknemer ON Werknemer.WerknemerID = Tickets.werknemerID,  
5  werkknemer Werknemer2 RIGHT JOIN AccountGegevens ON Werknemer2.Gebruikersnaam = AccountGegevens.Gebruikersnaam  
6  ORDER BY Tickets.Datum DESC;
```

Result Grid								
Filter Rows: <input type="text"/> Export: Wrap Cell Content:								
TicketNummer	Datum	Tier	Status	WerknemerID	Gebruikersnaam	Voornaam	Familienaam	
8	2022-12-10 00:00:00	3	Assigned en besproken	2	Michiel_Totaltickets	Michiel	Van Den Bempt	
7	2022-11-10 00:00:00	3	Assigned	1	Ewoud_Totaltickets	Ewoud	Forster	
4	2021-08-06 00:00:00	3	Afgewerkt	12	Francois_Totaltickets	Francois	Pater	
2	2021-01-11 00:00:00	1	Afgewerkt	10	Dirk_Totaltickets	Dirk	Deman	
10	2019-03-12 00:00:00	1	Afgewerkt	1	Ewoud_Totaltickets	Ewoud	Forster	

8. Kladnotities (Inzicht in ons project...)

8.1. Begin verslag

Samen gezeten en een bedrijf verzonnen, de naam van ons “bedrijf” is Total Tickets al onze stage bedrijven zijn klant van ons bedrijf.

Verzinnen wat de functie is van het bedrijf: Ons bedrijf krijgt de tickets van de klanten van andere bedrijven, en lossen deze op in verschillende levels.

Simpel ERD-model gemaakt om het concept te laten zien met de basis entiteiten. Hiervoor gebruikte we de applicatie LucidChart, een handig programma waarmee je makkelijk een ERD-model kunt maken.

Meer entiteiten aangemaakt voor uitbreiding en in klad gezet.

ERD-model aangemaakt met de entiteiten met lege attributen. Na dit hebben we al de entiteiten een relatie gegeven.

Op klad attributen verzonnen die passen bij hun entiteit, deze attributen zijn aangevuld bij hun passende entiteit, we probeerde zo veel mogelijk attributen te maken zodat we veel details hadden.

Klein probleempje gehad met het programma LucidChart waardoor het extra default tekst bij elk nieuw veld toevoegde. Geen echte oplossing gevonden, we moesten het bij hand verwijderen

Normalisatie proces gestart, tegelijk zijn we ons databank gestart in SQLworkbench

We hebben gevraagd aan onze leerkracht of we op een meer efficiënte manier bestanden kunnen delen en notificaties aan elkaar kunnen geven. We kregen dus ook een private groep waar we dit kunnen doen, hierdoor hebben we nu een privé kanaal in teams.

Aan de databank is gewerkt, maar er is een probleem met SQL Workbench waardoor het geen boolean wilt accepteren, door het te verwijderen terug te zetten werkte het terug.

Sebastiaan was ziek, hij heeft wel aan de normalisatie gewerkt, maar de communicatie was niet zo makkelijk, omdat hij thuis was.

Probleem van de SQL Workbench is terug, andere waarde aan boolean gegeven.

Ewoud aangereden net op de deadline van onze taak.

8.2. Algemeen

Idee uittekenen op papier

Kladnotities geschreven

ERD gemaakt zonder entiteiten

Relaties in het ERD model aangemaakt

Attributen in kladnotities en ERD-model gezet

Normalisatie

DB SQL workbench