**ORM**

**Wat is een ORM?**

Object-relational mapping is een library/tool dat wordt gebruikt voor het uitvoeren van SQL-query. Nu is het ook zo dat sommige ORM-tools de query voor je genereren. De ORM-tool zorgt ervoor dat het resultaat gelijk wordt omgezet naar de codetaal waarin je werkt. Dit waarschijnlijk als muziek in de oren. Maar hoe ziet het er daadwerkelijk uit, om je een beter beeld te geven laat ik hier een voorbeeld zien.  
Normaal zou je sql code er ongeveer zo uit zien:

Graphical user interface, text, application

Description automatically generated

Je maakt een object aan, je noteert je SQL query, leest de gegevens uit, zet deze over enzovoort.  
Wanneer je gebruik maakt van een ORM tool zal het er zo uit zien:



**Waarom ORM?**

Het eerste wat je waarschijnlijk opkomt is, dat het veel korter is. Dat klopt, met ORM is er veel meer geautomatiseerd waardoor je dus ook minder code nodig hebt. De belangrijkste kenmerken en gevolgen van ORM zijn:

* Het zorgt ervoor dat je DRY(don’t repeat yourself) werkt. Je schrijft je data model maar op een plek. Het aanpassen en herbruiken van je code is met ORM veel gemakelijker.
* Heel veel is voor je geautomatiseerd.
* Het forceert je om je code in MVC code te schrijven, wat je code er georganiseerd uit laat zien.
* Het voert SQL uit op een hoog niveau, dit kan fijn zijn wanneer je kennis ontbreekt in SQL.

Maar niks is perfect en daarom kijken we wat de voor- en nadelen van ORM zijn.

**De voordelen:**

* Je blijft in je eigen code schrijven (bv. C#).
* Het model is bijna niet verbonden met de rest van de applicatie waardoor aanpassingen makkelijker zijn.
* Makkelijk met OOP te gebruiken.

**De nadelen:**

* Je moet het leren gebruiken want het is geen kleine tool.
* Je moet het opzetten.
* Prestaties zijn oké, een geleerde SQL-master zal het wel beter doen dan een ORM tool.
* De SQL statements worden verborgen, waardoor je pas na het uitvoeren ziet wat er gebeurt.  
  Hierdoor kun je veel tijd kwijtraken als er een probleem is met je SQL.

Op basis van deze informatie ben ik gaan begrijpen wat ORM nou eigenlijk doet. Nu ben ik gaan kijken welke ORM-tool het beste past bij mij.  
Als eerst ben ik gaan kijken welke tools werken met Java, aangezien dat de taal wordt waarmee ik te werk ga.

Hibernate:

* Open source.
* Stabiel.
* Lightweight en krachtig.
* Support HQL.
* Snelle performance.
* Versimpeld joints.

TopLink:

* Heeft een GUI voor mapping.
* Betere mapping types.
* Makkelijk te integreren met EJB complaint container.

MyBatis:

* Gebruikt SQL i.p.v. HQL
* Flexibel
* Geeft tags i.p.v logic code
* Inline SQL support

**Conclusie keuze:**

Ik ben gaan kijken wat voor mij de belangrijkste kenmerken waren en welk het best bij mij past. Mijn eisen zat bij dat het open source is, omdat het dan gratis maar ook community supported is. Stabiliteit en lightweight gaan ook voor omdat mijn voorkeur gaat naar geen crashes en een vloeiend lopend proces. Daarom heb ik gekozen voor Hibernate, omdat deze net opgenoemde kenmerken het best aansluiten bij Hibernate.

**Bronnen:**

<https://www.javatpoint.com/orm-tools-in-java#:~:text=ORM%20stands%20for%20Object%20Relation,storage%20drivers%20in%20an%20API>.

<https://stackoverflow.com/questions/1279613/what-is-an-orm-how-does-it-work-and-how-should-i-use-one>

<https://stackoverflow.com/questions/49500230/what-is-the-difference-between-orm-and-dal#:~:text=The%20abstract%20concept%20of%20interacting,working%20with%20stored%20data%20easier>.

<https://www.youtube.com/watch?v=3EvhK7-DlZA>

<https://blog.bitsrc.io/what-is-an-orm-and-why-you-should-use-it-b2b6f75f5e2a>

<https://stackoverflow.com/questions/452385/what-java-orm-do-you-prefer-and-why>

**Asynchrone communicatie**

**Wat houdt asynchrone communicatie in?**

Het klinkt heel ingewikkeld maar dat is het eigenlijk helemaal niet. Asynchrone communicatie is communicatie waarbij je niet gelijk een antwoord verwacht of in andere woorden gezegd niet in realtime. Denk maar aan een whatsapp bericht of een email. Wanneer je zo’n bericht stuurt verwacht je niet dat het direct wordt beantwoord. Het tegenovergestelde hiervan is natuurlijk synchrone communicatie.

Dat is wanneer er direct wordt gecommuniceerd, zoals fysiek of digitaal met videobellen. Tegenwoordig wordt er veel meer asynchrone gecommuniceerd dan vroeger, dat komt omdat we nu veel verder zijn ontwikkeld en de technologieën ervoor hebben. En met de komst van de corona-crisis hebben we veel moeten omschakelen naar niet-fysieke contact, waardoor we een stuk meer asynchroon zijn gaan communiceren.

**Wanneer asynchrone communiceren en wanneer niet?**

Nu hebben beide zijn voor- en nadelen, en verschillen ze per situatie. Daarom kijken we waar beiden in uitsteken en minder zijn.

**De voordelen van Asynchrone communiceren:**

Wat fijn is aan asynchrone communicatie, is dat wanneer je een bericht binnenkrijgt je niet verplicht bent om direct te reageren. In plaats daarvan kun je eerst goed nadenken over wellicht een vraag die gesteld wordt, waardoor je een veel beter antwoordt geeft dan wanneer je dit direct zou moeten doen.   
  
Ook maakt het mogelijk om te werken op afstand. Wat nu in deze tijd ook veel meer wordt gedaan door middel van tools zoals Teams, Zoom, Trello, Whatsapp, etc... Zo breid je de mogelijk naar meer werknemer uit, en kun je op internationaal niveau te werk gaan.

Het geeft in de arbeidswereld meer vrijheid aan de werknemer, zo kan deze zijn tijd beter benutten. Zo scheelt de reistijd al en kan het beter gecombineerd worden met het persoonlijke leven.

**De nadelen van Asynchrone communiceren:**

Wanneer er een probleem is en er is sprake van tijdgebrek, kan het soms moeilijk zijn om te wachten op een antwoord. Daarom kan een telefoontje plegen of een videogesprek, een oplossing zijn voor het probleem. Dit kan in sommige situaties heel handig zijn.

A picture containing text, clipart

Description automatically generatedEen bericht bestaat enkel uit letters, en de gesproken emoties erachter zijn niet direct te achterhalen. Daardoor kun je soms een bericht verkeerd opnemen, en wordt het op de verkeerde manier opgepakt.

Zo kan de verstuurde het positief bedoelen, maar vat je het negatief op.

Zo zie je dat mensen deels thuiswerken en deels op kantoor. Daarom is het belangrijk om een goede balans te vinden tussen de twee.

**Bronnen:**

[https://www.globalization-partners.com/nl/blog/synchronous-vs-asynchronous-mastering-communication-in-the-remote-workplace/?utm\_source=Adwords&utm\_medium=cpc&utm\_campaign=Search\_|\_Netherlands\_-\_Dynamic\_[nl]&utm\_term=](https://www.globalization-partners.com/nl/blog/synchronous-vs-asynchronous-mastering-communication-in-the-remote-workplace/?utm_source=Adwords&utm_medium=cpc&utm_campaign=Search_|_Netherlands_-_Dynamic_%5bnl%5d&utm_term=)

<https://experience.dropbox.com/nl-nl/resources/asynchronous-communication>

<https://workninjas.nl/asynchrone-communicatie/>

<https://nozbe.com/nl/blog/asynchronous/>

https://onlinemarketingagency.nl/marketingtermen/asynchrone-communicatie/