## DBO

* Een productiemedewerker kan maar één teamleider hebben.
* Naam skillset en task moeten een naam en omschrijving hebben.
* Naam skillset en task moeten uniek zijn.
* Alle subtypes van employee gebruiken de primairy key van employee als hun eigen primairy key. Hierdoor heeft elke werknemer een unieke id. Dit is handig met de GetById functie in mijn repository. Alle werknemers komen in één lijst en als je er een selecteert, wordt die methode aangeroepen. Als niet elke werknemer, ongeacht subtype, geen unieke id heeft, moet ik ook speciferen voor welk type de GetById methode moet zoeken.
* Telefoonnummer wordt een nvarchar i.p.v. een int, want als een nummer begint met een 0, valt die weg. Dat wil ik niet.
* Op de tabel EmployeeProductionWorker heb ik een recursieve relatie. Als ik hieruit een record wil verwijderen die relaties met andere reccords in diezelfde tabel kan dat niet met een constraint op de foreign key met een ON DELETE SET NULL. Dit wordt niet ondersteund in Microsoft SQL-Server. Het is zodanig dichtgetimmerd. ([Source](https://devio.wordpress.com/2008/05/23/recursive-delete-in-sql-server/)) Dus wordt er een trigger (in stead of) gebruikt die eerst alle relaties verwijdert en daarna de records.

## EER-model

Een teamleider erft alles over van productiemedewerker, maar een teamleider kan momenteel nog geen teamleider hebben. Dit ondersteun ik nog niet vanwege de complexiteit. In de toekomst zou dit wel mogelijk kunnen zijn en dan is dit EER-model nog steeds correct.

## UML

Heb Task hernoemd naar ProductionTask, omdat ik erachterkwam dat er ook nog Threading.Task bestond. Dit heb ik gedaan om verwarring te verkomen.

T is generiek type