PDM

De taak:

1. Kopieer tabellen van Luca.
2. Voeg kolom toe: sleutels.
3. Voeg kolom toe: datatypen.
4. Bepaal regels voor update / delete (cascade?).
5. Bepaal 6 beperkingsregels (constraints).
6. Pas normalisatie toe, laat before & after zien. Geef uitleg.
   1. Leg uit waarom alle tabellen in 3NV staan.
7. PDM, misschien in PowerDesigner
8. Koppel bovenstaande eisen aan minimaal 3 use cases van het FO.

Om het gegevensmodel in SQL te kunnen implementeren moet je nadenken over de

fysieke implementatie. In dit hoofdstuk leg je de eisen vast waaraan de database moet

voldoen. Zorg ervoor dat deze duidelijk en ondubbelzinnig zijn verwoord.

Deze eisen bevatten in ieder geval:

- Primaire en secundaire sleutels

- Per attribuut: datatype passend bij een SQL server implementatie

- Regels ten aanzien van gewenst gedrag bij update / delete

- Zinvolle beperkingsregels -> formuleer zes logische beperkingsregels voor de

gegevens die je straks als ‘check constraints’ kunt implementeren.

Baseer je eisen op de usecases zoals gemaakt in het functioneel ontwerp, en wat je weet

over de verschillende entiteiten.

Je werkt het model uit in een PDM in Powerdesigner (of een eigen tool), waarbij je in het

rapport toelichting geeft op je keuzes, en zorgt dat de onderdelen hierboven allemaal

terugkomen.

Onderdeel van je toelichting op het fysieke model is ook een toelichting hoe je

normalisatie hebt toegepast. Je laat zien dat je de technieken van normalisatie beheerst

door drie voorbeelden te laten zien waar je redundantie hebt verwijderd uit het

gegevensmodel. Kijk naar elementen in de casusbeschrijving die duiden op niet-atomaire

gegevens. Maar laat ook een voorbeeld zien waar je al bij het maken van het PDM

redundantie hebt voorkomen. Toon van ieder voorbeeld dat je geeft 2 versies, waarin je

laat zien hoe de gegevens eerst waren, en wat de tabel(len) na normalisatie waren. (model

1NV/2NV/3NV)