

Van : Peter Cornelissen  
Voor : Semester BIS DT  
Betreft : Realisatie Database-applicatie GameParadise en rapportage  
Datum : 5 december 2017 – versie 2.2

### **Inleiding**

Voor het derde onderdeel ga je een deel van de applicatie voor GameParadise maken. De hele applicatie maken zou te veel werk zijn voor dit semester. Wat het belangrijkste is, zijn de rapportages. Daar zullen we ons in BP3 op concentreren. Daarnaast willen we ook graag zien dat je een zogenaamd Master-Detail-scherm kunt laten zien. Uiteraard is hier materiaal voor beschikbaar om dit te leren.

Voor de realisatie van de applicatie maken we gebruik van MS SQL server als RDMS. Hiermee zijn jullie al bekend. Wat nieuw is, is dat we met MS Access gaan werken om de applicatie aan de voorkant te maken. Daarvoor leggen we een koppeling tussen MS Access en MS SQL Server. Binnen Access kunnen we makkelijk schermen maken en binnen deze schermen kunnen we ook rapporten oproepen die ook met Access gemaakt kunnen worden.

*Nieuw dit jaar is dat studenten de vrijheid hebben om ook een andere Front-end te gebruiken dan Access. Diegene die bedreven is in een andere programmeertaal mag dit gebruiken. De database moet wel MS SQL Server 2016 zijn.*

*Echter: ondersteuning wordt geboden bij het gebruik van MS Access. De docent kan geen support geven op alle individueel gekozen andere front-end ontwikkeltaalen en -omgevingen.*

LET OP! De SQL-regels voor de rapportage moeten worden gemaakt in MS SQL Server. MS Access dient slechts als eenvoudige front-end-ontwikkeltool.

Voor de implementatie van het begin van de beveiliging bouwen we ook een menu. Afhankelijk van wie er “inlogt” zal dit menu toegang geven tot andere schermen.

### **Rapport!**

Tot slot moet er een beperkt rapport worden gemaakt waarin je je gemaakte keuzes verantwoord. Het beste kun je tijdens het realiseren notities maken van de keuzes die je op dat moment maakt. Dan is het daarna vrij eenvoudig om deze netjes in een rapport te zetten.

### **OPDRACHTEN**

Hieronder vind je opdrachten waarin je in stappen de applicatie voor GameParadise maakt. Per opdracht is vermeld of het een groepsopdracht (2 of 3 studenten) of een individuele opdracht is.

#### **Opdracht 1: Optimaliseer je database en voeg de voorbeelddata toe (GROEP)**

- A. In het vorige beroepsproduct heb je een database moeten maken. Bij de documenten vind je een uitwerking van het DDL-script (GameParadise-DDL.sql). Vergelijk jullie uitwerking met de uitwerking zoals deze wordt aangereikt. Verklaar de verschillen en maak een onderbouwde keuze voor een van de twee oplossingen (die van jullie zelf of die van de uitwerking). De

keuzes en de onderbouwing neem je op in je rapport. Het rapport wordt individueel ingeleverd, voor de groepsopdrachten zal dus de inhoud van het rapport hetzelfde zijn.

BEWAAR JE EIGEN DDL-SCRIPT EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN. DE NAAM VAN DE FILE GRAAG ZO SAMENSTELLEN: **GP-DDL-SCRIPT-[NAAM STUDENT].SQL**. verzenden in een RAR- of ZIP-formaat mag ook.

- B. Bij de bestanden vind je een INSERT-script (GameParadise-INSERT.sql) waarmee een grote hoeveelheid data als testdata in de database gezet kan worden. Zorg ervoor dat deze data in je database komt. Hiervoor moet je mogelijk het INSERT-SCRIPT aanpassen. ***De aanpassingen die je hebt moeten doen moet je ook opnemen en toelichten in je rapport.***

BEWAAR JE EIGEN INSERT-SCRIPT EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN. TEGELIJK IN. DE NAAM VAN DE FILE GRAAG ZO SAMENSTELLEN: **GP-DML-SCRIPT-[NAAM STUDENT].SQL**.

### **Opdracht 2: CONSTRAINTS TOEVOEGEN (GROEP)**

Probeer de volgende constraints (business rules) te definiëren in de CREATE TABLE statements:

- Het geslacht van een klant moet 'M' (man) of 'V' (vrouw) zijn.
- De einddatum van een verhuur moet later zijn dan de begindatum van een verhuur.
- De betaaldatum (startdatum) van een schade moet eerder of gelijk zijn t.o.v. de gereeddatum van een schade.
- Het adres van een klant moet uniek zijn.
- Probeer nog minimaal 3 voor de hand liggende (en dus realistische) constraints te vinden uit de casus, beschrijf deze en realiseer deze indien mogelijk.

BEWAAR HET AANGEPASTE CREATESCRIPT ONDER EEN ANDERE NAAM EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN. DE NAAM VAN DE FILE GRAAG ZO SAMENSTELLEN: **GP-DDL-SCRIPT-CONSTRAINTS-[NAAM STUDENT].SQL**.

### **Opdracht 3: CASCADING (GROEP)**

Pas de DDL-statements aan zodat ook de update- en deleteregels (ON UPDATE ....., ON DELETE .....) worden geïmplementeerd.

BEWAAR HET AANGEPASTE CREATESCRIPT ONDER EEN ANDERE NAAM EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN. DE NAAM VAN DE FILE GRAAG ZO SAMENSTELLEN: **GP-DDL-UPDATE-DELETE-SCRIPT-[NAAM STUDENT].SQL**.

### **Opdracht 4: TESTEN (GROEP)**

Test alle constraints die in 2 en 3 gecreëerd zijn. Toon door middel van INSERT-statements voor het toevoegen van voorbeeldpopulatie (goede populatie en tegenvoorbeelden) aan dat de geïmplementeerde constraints uit opdrachten 2 en 3 correct geïmplementeerd zijn.

BEWAAR DIT SCRIPT MET GOEDE POPULATIE EN TEGENPOPULATIE ALS **GP\_TESTSCRIPT-[NAAM STUDENT].SQL**.

### **Opdracht 5: Koppeling tussen MS SQL Server en MS Access (GROEP)**

Maak een koppeling op basis van een ODBC-driver tussen MS SQL Server 2016 en MS Access 2016. Hoe dit moet, kun je lezen in de reader MS ACCESS V4.0 hoofdstuk 3.

Mocht je een andere Front-end-tool willen gebruiken, zorg dan dat er tussen de door jou gekozen tool en MS SQL SERVER een koppeling wordt gelegd en leg in de verantwoording (Verslag) uit hoe je dit gerealiseerd hebt.

### **Opdracht 6: Maak queries voor het genereren van overzichten (GROEP)**

Maak SQL Select-queries voor onderstaande informatiebehoeften. Dit dient twee doelen. Op deze manier leer je de database goed kennen. Een deel van de queries kun je straks als basis voor de realisatie van de applicatie gebruiken, bijv. als recordbron van een formulier of rapport of als rijbron van een combobox of listbox. Maak queries voor de volgende informatiebehoefte; toon daarbij de kolommen die tussen de blokhaken staan:

- A. Alle artikelen in voorraad gesorteerd naar soort, merk en type. Dus die artikelen die op het moment niet verkocht of verhuurd zijn.
- B. Alle Consoles die in 2016 schade hebben gehad.
- C. Alle klanten die momenteel een Console hebben gehuurd. Welke klant heeft welke console gehuurd.
- D. De artikelen die verhuurd zijn in mei 2015.
- E. De huurovereenkomsten waarin een Playstation is gehuurd.
- F. Alle klanten die in 2017 iets gekocht of gehuurd hebben. Deze willen we namelijk een kerstkaart sturen.

BEWAAR DIT SCRIPT MET QUERIES **GP\_QUERY-SCRIPT-[NAAM STUDENT].SQL**

### **Opdracht 7: Maak een View voor deze informatiebehoefte (INDIVIDUEEL!)**

- A. De omzet van de maand januari in het de jaar 2015.
- B. Het meest gehuurde Spel.
- C. De huurovereenkomst met de hoogste omzet.
- D. De totale schade in het jaar 2016.
- E. De consoles met de meeste schade.

BEWAAR DIT SCRIPT MET VIEWS ALS **GP\_VIEW-SCRIPT-[NAAM STUDENT].SQL**

### **Opdracht 8: Implementeer een ontworpen scherm in Access met een Master-Detail-oplossing (INDIVIDUEEL!).**

In BeroepsProduct 1 (BP1) heb je aan aantal schermen ontworpen. Kies een van deze ontworpen schermen uit om te implementeren als formulier in Access. Om te leren hoe dit moet kun je de volgende video op PluralSight bekijken: [PluralSight-Video]

**BELANGRIJK!** Je moet wel een ontworpen scherm uitkiezen waarbij een master-detail-scherm kunt laten zien.

### **Opdracht 9: Bouw Rapporten in Access (INDIVIDUEEL!).**

Bouw voor de volgende informatiebehoefte rapporten in Access:

- A. Overzicht van artikelen die momenteel verhuurd zijn.
- B. Lijst met Spellen die nog niet verhuurd zijn in 2016.
- C. Omzetten van iedere maand van de jaren 2012 tot en met 2017.
- D. Overzicht van de top 10 van populairste spellen.
- E. Consoles die in reparatie staan.
- F. Omzetten per klant in een jaar, gesorteerd van hoog naar laag. Verdeel de klanten in **Gold**, **Silver**, **Bronze** zoals beschreven in de casus.
- G. Inkoophoeveelheid (aantal artikelen) en totaalbedrag per jaar.

Om deze rapporten te maken **moet** je eerst views ontwikkelen in MS SQL SERVER. Dit heeft als voordeel dat je in Access alleen de View hoeft aan te roepen.

BEWAAR DE SQL-STATEMENTS VOOR DE RAPPORTEN-VIEWS **GP\_RAPPORTEN-VIEWS-SCRIPT-[NAAM STUDENT].SQL**

### **Opdracht 10: Beveiliging (INDIVIDUEEL!).**

De rapporten uit opdracht 9 mogen niet door iedereen ingezien worden. De volgende rechten zijn verdeeld:

RAPPORT A: MEDEWERKER, FILIAALMANAGER, DIRECTIE, VOORRAADBEHEERDER

RAPPORT B: VOORRAADBEHEERDER, FILIAALMANAGER, DIRECTIE

RAPPORT C: DIRECTIE

RAPPORT D: VOORRAADBEHEERDER, FILIAALMANAGER, DIRECTIE

RAPPORT E: VOORRAADBEHEERDER, REPARATEUR, MEDEWERKER, FILIAALMANAGER

RAPPORT F: FILIAALMANAGER, DIRECTIE

RAPPORT G: DIRECTIE

Bouw een Menu-Formulier in Access (of je eigen gekozen Front-End Tool) waarbij je virtueel (met knoppen) kunt inloggen als REPARATEUR, MEDEWERKER, VOORRAADBEHEERDER, FILIAALMANAGER of DIRECTIE. Aan de hand van de rol waarmee ingelogd worden, wordt er een ander scherm geopend met daarbij toegang tot de rapporten waar men rechten op heeft.

Dit is slechts een begin van de implementatie van gebruikersrechten. Natuurlijk is dit niet veilig en zal dit in een professionele organisatie verder moeten worden afgeschermd. Dit is echter wel het begin van de toepassing van beveiliging binnen een applicatie.

BEWAAR JE ACCESS-APPLICATIE EN STUUR DEZE MEE! OF BEWAAR DE APPLICATIE, GEBOUWD IN JE EIGEN FRONT-END-TOOL EN STUUR DEZE MEE. **GP\_APPLICATION-[NAAM STUDENT].ACCDB (of andere extensie als je gekozen hebt voor een andere front-end-tool.**

### **Opdracht 11: volgende bladzijde!**

**Opdracht 11: Samenstellen rapport (INDIVIDUEEL!).**

Maak een beknopt rapport van de resultaten van je werkzaamheden. Daarbij gaat het vooral om de uitwerkingen met daarbij de toelichting op de keuzes die je gemaakt hebt.

Het rapport moet van dezelfde kwaliteit zijn, zoals dat omschreven is in BP2. De kwaliteit dient gecontroleerd te worden met de ICA-controlekaart.

Onderdelen die moeten terugkomen in het rapport:

- a. Verklaring van de verschillen tussen je eigen CREATE SCRIPT en de aangereikte uitwerking (opdracht 1) en een onderbouwde keuze voor een van de twee oplossingen.
- b. Toegevoegde constraints en toelichting (opdracht 2)
- c. Cascading, en toelichting (opdracht 3)
- d. Queries voor overzichten en toelichting (opdracht 6)
- e. Views voor informatiebehoefte (opdracht 7)
- f. Keuze voor geïmplementeerd scherm, waarom en onderbouwing (opdracht 8)
- g. Views voor de rapportage (opdracht 9)
- h. Keuzes en aanpak bij opdracht 10.
- i. Reflectie; wat vond je het lastigste en waarom? Wat zou je anders doen een volgende keer?

**INLEVEREN:**

**UPLOAD JE COMPLETE UITWERKINGEN MET ALLE BESTANDEN IN EEN ZIP-BESTAND. GP\_COMPLEET-[NAAM STUDENT].ZIP. DOE DIT VOOR DE AANGEGEVEN DATUM EN TIJDSSTIP.**