

Van : Peter Cornelissen
Voor : Semester BIS DT
Betreft : Realisatie Database-applicatie Biker en rapportage
Datum : 4 december 2018 – versie 2.1

Inleiding

Voor het derde onderdeel gaan jullie een deel van de applicatie voor Biker maken. De hele applicatie maken zou te veel werk zijn voor deze module. Wat het belangrijkste is zijn de rapportages. Daar zullen we ons in BP3 op concentreren. Daarnaast willen we ook graag zien dat jullie een zogenaamd Master-Detail-scherm kunnen 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 (of een andere omgeving, zie volgende alinea) gaan werken om de applicatie aan de voorkant (front end) 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.

Studenten hebben de vrijheid om ook een andere Front-end-ontwikkelomgeving te gebruiken dan MS Access. Diegene die bedreven is in een andere programmeertaal, mag dit gebruiken. De database moet wel MS SQL Server 2017 of 2016 zijn.

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

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.

Samenwerking

Alle opdrachten worden in groepen uitgevoerd. Het beste kunnen jullie de groep handhaven waarmee jullie ook in het vorige deel het ontwerp en de database hebben gemaakt. Mocht dat niet wenselijk zijn dan kunnen jullie ook daarin schuiven. De samenstelling van de groep dienen jullie wel door te geven aan jullie docent in de eerste week van dit derde deel van deze module (dus eigenlijk nu direct nadat jullie dit gelezen hebben).

Voor het assessment moeten alle studenten kennis hebben van alle onderdelen van dit beroepsproduct. Werk verdelen is aan te bevelen maar zorg wel dat jullie het werk van de ander begrijpen, kunnen uitleggen en reproduceren.

Rapport

Er wordt een rapport gemaakt waarin jullie, jullie gemaakte keuzes verantwoorden. Het beste kunnen jullie tijdens het realiseren notities maken van de keuzes die jullie op dat moment maken. Dan is het daarna vrij makkelijk om deze netjes in een rapport te zetten.

OPDRACHTEN

Hieronder vinden jullie opdrachten waarin jullie in stappen de applicatie voor Biker maakt. Elke student in de groep zou in staat moeten zijn om de opdrachten te maken. Dat wil niet zeggen dat ook elke student van de groep alle opdrachten maakt. Uiteraard mogen jullie het werk verdelen.

Opdracht 1: Optimaliseer jullie database en voeg de voorbeelddata toe.

- A. In het vorige beroepsproduct hebben jullie een database moeten maken. Bij de documenten vinden jullie een uitwerking van het DDL-script (Biker-DDL-script.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 nemen jullie op in jullie rapport. Het rapport en bijlagen wordt individueel ingeleverd (via iSAS), voor de groepsopdrachten zal dus de inhoud van het rapport hetzelfde zijn.

BEWAAR JULLIE EIGEN DDL-SCRIPT EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN.

- B. Bij de bestanden vinden jullie een INSERT-script (Biker-DML-script.sql) waarmee een grote hoeveelheid data als testdata in de database gezet kan worden. Zorg ervoor dat deze data in jullie database komt. Hiervoor moeten jullie mogelijk het INSERT-SCRIPT aanpassen. De aanpassingen die jullie hebben moeten doen moeten jullie ook opnemen en toelichten in jullie rapport.

BEWAAR JULLIE EIGEN INSERT-SCRIPT EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN.

Opdracht 2: CONSTRAINTS TOEVOEGEN

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.
- Het Einddatum van een verhuur moet later zijn dan de begindatum van een verhuur.
- De betaaldatum van een schade moet later zijn dan een begindatum van een schade.
- Het e-mailadres van een klant moet uniek zijn.
- Een begindatum van een verzekering moet gelijk of eerder zijn aan de begindatum van een verhuur.
- De dagprijs van een fiets moet lager zijn dan de nieuwwaarde.
- Een medewerker en een klanten moeten minstens 18 jaar oud zijn.
- Er mag alleen maar een accessoire verhuurd worden als er ook fiets gehuurd wordt.
- Er mogen per gehuurde fiets maar 2 accessoires gehuurd worden.
- Er mag alleen een schade aan een fiets zijn als deze ook door dezelfde klant gehuurd is in de periode voorafgaand aan de schademelding.
- Probeer nog minimaal 3 voor de hand liggende (en dus realistische) constraints te vinden, beschrijf deze en realiseer deze indien mogelijk.

BEWAAR HET AANGEPASTE CREATESCRIPPT ONDER EEN ANDERE NAAM EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN.

Opdracht 3: CASCADING

Pas de DDL-statements in jullie eigen script (of het aangeleverde script) aan zodat ook de update- en deleteregels (ON UPDATE, ON DELETE) worden geïmplementeerd. Beargumenteer de toevoegingen of aanpassingen.

BEWAAR HET AANGEPASTE CREATESCRIPT ONDER EEN ANDERE NAAM EN LEVER DEZE AAN HET EIND MET ALLE ANDERE BESTANDEN TEGELIJK IN.

Opdracht 4: TESTEN

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
BIKER_TESTSCRIPT.SQL

Opdracht 5: Koppeling tussen MS SQL Server en MS Access

Maak een koppeling op basis van een ODBC-driver tussen MS SQL Server 2017 of 2016 en MS Access 2016. Hoe dit moet, kun je lezen in de reader MS ACCESS V4.0 hoofdstuk 3.

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

Opdracht 6: Maak queries voor het genereren van overzichten

Maak SQL Select-queries voor onderstaande informatiebehoeften. Dit dient twee doelen. Op deze manier leren jullie de database goed kennen. En een deel van de queries kunnen jullie 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:

- A. Alle fietsen gesorteerd na soort [elektrisch, dames/heren]
- B. Alle fietsen die in 2012 schade hebben gehad.
- C. Alle klanten die momenteel een fiets hebben gehuurd.
- D. De fietsen en accessoires die gehuurd zijn onder huurovereenkomst 15.
- E. De huurovereenkomsten waarin een Gazelle fiets is gehuurd.
- F. Alle klanten met nog niet betaalde rekeningen
- G. Aantal gereserveerde, geannuleerde en betaalde huurovereenkomsten

BEWAAR DIT SCRIPT MET QUERIES BIKER_QUERY-SCRIPT.SQL

Opdracht 7: Maak een View voor deze informatiebehoefte

- A. De omzet van de maand januari in het de jaar 2012.
- B. De meest gehuurde fiets (Top 100).
- C. Het meest gehuurde accessoire (Top 100).
- D. De huurovereenkomst met de hoogste omzet.
- E. De schade in het jaar 2016
- F. De fiets met de meeste schade
- G. De klant met de hoogste omzet van 2017

BEWAAR DIT SCRIPT MET VIEWS ALS BIKER_VIEW-SCRIPT.SQL

Opdracht 8: Implementeer een ontworpen scherm in MS Access (of een andere ontwikkelomgeving) met een Master-Detail-oplossing

In BeroepsProduct 1 (BP1) hebben jullie aan aantal schermen ontworpen. Als het goed is ook een scherm om offertes samen te stellen. Jullie gaan het samenstellen van een offerte implementeren als formulier in MS Access (of een andere ontwikkelomgeving). Om te leren hoe dit moet kunnen jullie de video op PluralSight bekijken: <https://app.pluralsight.com/player?course=microsoft-access-2016&author=patricia-ciszek&name=microsoft-access-2016-m4&clip=0&mode=live>

BELANGRIJK! In deze implementatie moet een master-detail-scherm zichtbaar worden. Een voorbeeld van een mogelijke oplossing:

1. Je laat als eerste een lijst met klanten zien
2. Wanneer je door een klanten loopt met de pijltjestoetsen of muis worden elders op het scherm de door die klant afgesloten huurovereenkomsten zichtbaar met eventueel in die huurovereenkomst gehuurde fietsen en accessoires.
3. Er is een knop “maak nieuwe offerte”, wanneer je op de knop drukt wordt er voor de klant die je op dat moment hebt geselecteerd, een nieuw scherm geopend waarin je:
 - a. Fietsen selecteert
 - b. Accessoires selecteert
 - c. De begindatum vastlegt (in de toekomst)
 - d. En de einddatum vastlegt.
4. De totaalprijs moet berekend worden en ook de borg. Daarbij moet rekening gehouden worden of de geselecteerde klant een “trouwe” klant is. Als dat zou is moet dit zichtbaar worden op de offerte.
5. Wanneer de offerte samengesteld is en op de knop “verstuur” gedrukt wordt, moet er een rapport gemaakt worden waarop alle gegevens voor de offerte voor die klant zichtbaar zijn.

Het kan wellicht handig zijn om hetgeen er uitgewerkt wordt in opgave 9 ook voor deze opdracht te gebruiken.

In opdracht 10 ga je ook een beperkte menu-structuur bouwen waar vanuit je dit scherm kunt aanroepen wanneer je met de juiste rol inlogt.

!! In de applicatie die jullie gaan bouwen passen jullie geen SQL toe, behalve het aanroepen van views met eventueel filtering!!

Opdracht 9: Bouw Rapporten in Access.

Bouw voor de volgende informatiebehoefte rapporten in Access:

- A. Overzicht van fietsen die momenteel verhuurd zijn.
- B. Overzicht van accessoires die momenteel verhuurd zijn.
- C. Lijst met fietsen die nog niet verhuurd zijn in 2018
- D. Lijst met accessoires die de laatste 2 jaar niet verhuurd zijn. Deze gaan ze op marktplaats verkopen.
- E. Omzetten van iedere maand van de jaren 2010 tot en met 2018
- F. Overzicht van de top 10 van populairste fietsen (vaakst verhuurd).
- G. Overzicht van de top 100 beste klanten van het afgelopen jaar, die gaan ze een kerstkaart sturen. Beste klanten zijn die het meeste besteed hebben.
- H. Overzicht van de top 10 klanten die het vaakst na een verhuur de fietsen terugbrengen met schade. Daarbij willen ze weten hoe vaak ze in totaal gehuurd hebben en hoeveel keer er schade is geweest. Dan weet je ook het percentage schadegevallen per klant, wat ook in het rapport kan worden opgenomen.
- I. Lijst met fietsen met het meeste aantal schades (op basis van aantal en op basis van schadebedrag).
- J. Percentage geannuleerde fietsverhuur-aanvragen in 2015 t/m 2018.
- K. Trouwe klanten van 2017 en 2018.
- L. Totaal betaalde borg in de jaren 2015 t/m 2018.

Om deze rapporten te maken moeten jullie eerst views ontwikkelen in MS SQL SERVER. Dit heeft als voordeel dat jullie in Access alleen de View hoeven aan te roepen.

!! In de applicatie die jullie gaan bouwen passen jullie geen SQL toe, behalve het aanroepen van views met eventueel filtering!!

BEWAAR DE SQL-STATEMENTS VOOR DE RAPPORTEN-VIEWS BIKER_RAPPORTEN-VIEWS-SCRIPT.SQL

Opdracht 10: Beveiliging.

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

RAPPORT A: VERKOPER, DIRECTIE	RAPPORT G: VERKOPER, DIRECTIE
RAPPORT B: VERKOPER, MONTEUR, DIRECTIE	RAPPORT H: DIRECTIE
RAPPORT C: DIRECTIE	RAPPORT I: MONTEUR
RAPPORT D: VERKOPER, DIRECTIE	RAPPORT J: DIRECTIE
RAPPORT E: VERKOPER, DIRECTIE	RAPPORT K: VERKOPER, DIRECTIE
RAPPORT F: VERKOPER, MONTEUR, DIRECTIE	RAPPORT L: DIRECTIE

Bouw een Menu-Formulier in Access waarbij jullie inloggen met de inlognaam en het wachtwoord van een medewerker. Deze gegevens worden gecontroleerd in de database en afhankelijk van de rol(len) die de medewerker heeft, wordt er een menuscherm geopend met daarbij toegang tot de rapporten waar deze persoon 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 JULLIE APPLICATIE EN STUUR DEZE MEE! Tijdens het assessment laten jullie op jullie eigen computer de applicatie zien. De docent installeert niet alle applicaties op zijn eigen computer.

Opdracht 11: Samenstellen rapport

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

Het rapport moet van dezelfde kwaliteit zijn, zoals dat omschreven is in BP2.

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 offerte-samenstellen-scherm, waarom en onderbouwing (opdracht 8)
- g. Views voor de rapportage (opdracht 9)
- h. Keuzes en aanpak bij opdracht 10.
- i. Reflectie per groepslid; wat vond je het lastigste en waarom? Wat zou je anders doen een volgende keer?

INLEVEREN:

UPLOAD JULLIE COMPLETE UITWERKINGEN MET ALLE BESTANDEN IN EEN ZIP-BESTAND. DOE DIT VOOR DE AANGEGEVEN DATUM EN TIJDSSTIP. DUS ELK INDIVIDUEEL GROEPSLID DOET.