

# Programmeren met databanken volgens het lagenmodel in C# met Visual Studio en MySQL

HA-2267-04

Informaticabeheer  
Bert Cauwenberg en Lieven Pauwels  
Werkgroep Handel  
2017





# Programmeren met databanken volgens het lagenmodel in C# met Visual Studio en MySQL

---

HA-2267-04

Informaticabeheer  
Bert Cauwenberg en Lieven Pauwels  
Werkgroep Handel  
2017



---

# Inhoud

<b>1</b>	<b>Inleiding.....</b>	<b>5</b>
1.1	Data(banken) in het lagenmodel.....	5
1.2	Terminologie en afkortingen .....	6
1.2.1	Databanken .....	6
1.2.2	Enkele afkortingen .....	6
1.3	Situering in de leerstof programmeren .....	7
1.4	Didactische opbouw van de cursus .....	8
1.4.1	Einddoel van deze cursus.....	8
1.4.2	Didactische opbouw doorheen de hoofdstukken .....	8
1.4.3	Typografische afspraken en schematisering .....	9
1.4.4	Gebruik van class libraries uit HA-2265 en HA-2266.....	9
<b>2</b>	<b>Algemene principes en werkwijzen .....</b>	<b>11</b>
2.1	Met een databank verbinden vanuit programmacode .....	11
2.1.0	Inleiding .....	11
2.1.1	Een MySQL-connectie maken vanuit C#-code .....	12
2.1.2	Een opvraagquery uitvoeren vanuit C#-code .....	12
2.1.3	Een toevoegquery uitvoeren vanuit C#-code .....	14
2.1.4	Een updatequery uitvoeren vanuit C#-code .....	14
2.1.5	Een verwijderquery uitvoeren vanuit C#-code .....	15
2.2	Basisvoorbeeld .....	16
2.2.1	Punten tonen .....	16
2.2.2	Met OOP .....	18
2.2.3	In drie aparte lagen .....	20
2.3	Centrale probleemstelling: Punten .....	23
2.3.1	Omschrijving.....	23
2.3.2	Verschillende aanpakken .....	24
2.4	De programmacode uitwerken volgens drie lagen .....	25
2.4.1	Probleemanalyse.....	25
2.4.2	Structuur en naamgeving voor de databank .....	29
2.4.3	Voorbeeldcode .....	30
2.4.4	Oefening.....	35
2.5	Het gebruik van een repository .....	36
2.5.0	Inleiding .....	36
2.5.1	Probleemanalyse.....	36
2.5.2	Voorbeeldcode .....	38
2.5.3	Oefeningen.....	40
2.6	Het gebruik van een mapper .....	41
2.6.1	Probleemanalyse.....	41
2.6.2	Voorbeeldcode .....	44
2.6.3	Oefeningen.....	45
2.7	Stappenplan en aandachtspunten .....	46
2.7.1	Stappenplan .....	46
2.7.2	Schematische weergaven.....	47
2.7.3	Aandachtspunten i.v.m. de werkwijze .....	49
2.7.4	Tips aangaande de Visual Studio-solutions .....	52
<b>3</b>	<b>Voorbeeld: Namen van leerlingen opslaan.....</b>	<b>55</b>
3.1	Probleemstelling .....	55
3.2	Ontwerp UI en user interaction .....	55
3.3	Klassendiagram.....	56
3.4	Structuur van de databank.....	58
3.5	Structuur van de solution .....	58
3.6	Voorbeeldcode: Namen .....	59
3.6.1	Business controller .....	59
3.6.2	PersoonRepository .....	59
3.6.3	Persistence Controller .....	60
3.6.4	PersoonMapper .....	60

3.7	Oefeningen .....	61
3.8	Uitbreidingen .....	62
3.8.1	De DataSource-eigenschap van een ListBox .....	62
3.8.2	De gegevens gesorteerd weergeven .....	63
3.8.3	Een geselecteerd gegeven verwijderen .....	65
3.8.4	Een geselecteerd gegeven wijzigen .....	67
<b>4</b>	<b>Voorbeeld: DiceGame met een score-databank.....</b>	<b>69</b>
4.1	Uitgangspunten .....	69
4.2	Klassendiagram .....	70
4.3	Samenwerking tussen business controller, repository en persistence controller .....	71
4.4	Structuur van de databank .....	71
4.5	Structuur van de solution .....	71
4.6	Voorbeeldcode: score-databank voor DiceGame .....	72
4.6.1	Business controller .....	72
4.6.2	SpelerRepository .....	72
4.6.3	Persistence Controller .....	73
4.6.4	SpelerMapper .....	73
4.7	Oefeningen .....	74
<b>5</b>	<b>Aanvullingen en uitbreiding .....</b>	<b>75</b>
5.1	De connection string niet hard coded implementeren .....	75
5.2	Levensduur van een mapper .....	77
5.3	Een repository als singleton .....	78
5.4	Database-onafhankelijk programmeren.....	79
	<b>Bijlage A: Terminologie i.v.m. databanken .....</b>	<b>83</b>
A.1	Enkele datamodellen .....	84
A.1.1	Plat datamodel .....	84
A.1.2	Hiërarchisch datamodel.....	84
A.1.3	Netwerk-datamodel .....	85
A.1.4	Relationeel datamodel .....	85
A.1.5	Objectgeoriënteerd datamodel .....	86
A.2	Enkele DBMS'en .....	87
A.2.1	MS Access.....	87
A.2.2	MS Access SQL .....	87
A.2.3	MS SQL Server.....	87
A.2.4	MySQL Server .....	88
A.2.5	FireBird en JayBird .....	88
A.2.6	Oracle Database.....	88
A.3	Databanken en programmeren .....	89
A.3.1	ODBC en Microsoft-API's .....	89
A.3.2	Repositories .....	92
A.3.3	Mappers.....	92
A.3.4	Samenvatting .....	93
	<b>Bijlage B: Gebruik van MySQL en ADO.NET met C# .....</b>	<b>95</b>
B.1	Basisbegrippen i.v.m. ADO.NET .....	95
B.1.1	ADO.NET .....	95
B.1.2	Connected en disconnected.....	95
B.1.3	MySQL en .NET .....	96
B.2	MySQL-databanken gebruiken in Visual Studio .....	97
B.2.1	Een reference toevoegen aan de MySql.Data-library .....	97
B.2.2	Verbinden met een MySQL-databank .....	98
B.2.3	Een MySQL-databank aanmaken .....	101
B.2.4	Toevoegen van tabellen .....	103
B.2.5	Een tabel voorzien van een primaire sleutel.....	104
B.2.6	Gegevens opvragen en toevoegen in de GUI van Visual Studio .....	106
B.2.7	Een bestaand MySQL-databank-bestand toevoegen.....	107
B.2.8	Connection strings .....	108

<b>Bijlage C: Gegevens opslaan in een bestand .....</b>	<b>109</b>	
C.1	Probleemstelling: score-functionaliteit voor DiceGame .....	109
C.2	Probleemanalyse .....	112
C.2.1	Bijhouden en tellen van de score .....	112
C.2.2	Aanpassing interface .....	112
C.2.3	Opslaan in een globale variabele .....	113
C.2.4	Opslaan in een bestand .....	113
C.2.5	Opslaan in een databank .....	114
C.3	Implementatie .....	116
C.3.1	Structuur voor persistence code in de solution .....	116
C.3.2	Klassendiagram .....	118
C.3.3	Gegevens opslaan in plain text (TXT) .....	119
C.3.4	Voorbeeldcode .....	119
C.3.5	Het gebruik van StreamWriter .....	120
C.3.6	Samenvatting en aandachtspunten .....	120
C.4	Oefeningen en uitbreiding .....	122
C.4.1	Oefeningen .....	122
C.4.2	Uitbreiding: ander formaat om gegevens op te slaan (CSV) .....	125
C.4.3	Besluit .....	127

