|  |
| --- |
| Opdracht 3 - Performance |
| Verslag |
| Analyse 8 - Advanced Databases (2014-2015) |

|  |
| --- |
|  |
|  |

Namen: HoYe Lam, Rinesh Ramadhin

Studentnummer: 0876814, 0882447

Klas: INF2D

Vak: INFANL01-8

Opdracht: Performance

Datum: 27 – 05 - 2015

Inhoud

[Simulatie: Gebruikers die data invoeren 2](#_Toc421039071)

[Simulatie: gebruikers die data opvragen 8](#_Toc421039072)

[Simulatie: Indexen 11](#_Toc421039073)

# Simulatie: Gebruikers die data invoeren

Hieronder vindt U de JAVA code die we hebben geschreven om gelijktijdig meerdere data in te voeren in de POSTGRES database. Er wordt per iteratie geprint hoelang elke thread erover doet.

We hebben in elke thread een aantal INSERTS uitgevoerd. De gemeten tijd per iteratie ligt tussen de 0 en 4 milliseconden.

De voornamen van de studenten zijn random gegenereerd, zodat we unieke waardes hebben voor gebruik bij de index simulatie.

Uiteindelijk wordt de tijd per iteratie uitgeprint. Aan het eind van elke thread wordt de gemiddelde tijd van de iteraties uitgeprint.

**import** java**.**sql**.**Connection**;**

**import** java**.**sql**.**DriverManager**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**sql**.**Statement**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**Random**;**

public class DataInvoegen **{**

// decladerdeer variabele

//Connectie Variables

public static String url **=** "jdbc:postgresql://127.0.0.1:5432/postgres"**;**

public static String username **=** "postgres"**;**

public static String password **=** "hoye"**;**

public static Random rand **=** **new** Random**();**

public static void main**(**String**[]** args**)** **{**

// Maak en start thread 1

**new** Thread**(new** Runnable**()** **{**

@Override

public void run**()** **{**

Connection connection **=** connect**();**

// Decladeer Thread 1 variables

List**<**String**>** groep **=** **new** ArrayList**<**String**>();**

List**<**String**>** module **=** **new** ArrayList**<**String**>();**

List**<**Long**>** totale\_duur\_thread1 **=** **new** ArrayList**<**Long**>();**

long begintijd**;**

long eindtijd**;**

long duurInms**;**

// Onze eerste klas en module

Statement st1**;**

**try** **{**

st1 **=** **(**Statement**)** connection**.**createStatement**();**

st1**.**executeUpdate**(**"INSERT INTO Groep "

**+** "VALUES ('INF1D', '2015-05-05', '2016-01-01')"**);**

groep**.**add**(**"INF1D"**);**

st1**.**executeUpdate**(**"INSERT INTO Cursussen "

**+** "VALUES ('INFANL-01','TJANG','analyse','2015-05-05', '2016-01-01')"**);**

module**.**add**(**"INFANL-01"**);**

**}** **catch** **(**SQLException e2**)** **{**

// TODO Auto-generated catch block

e2**.**printStackTrace**();**

**}**

// Begin 600 iteraties van student en klas toevoegen

**for** **(**int i **=** 0**;** i **<** 600**;** i**++)** **{**

// sla begin tijd op

begintijd **=** System**.**currentTimeMillis**();**

// random string

char**[]** chars **=** "abcdefghijklmnopqrstuvwxyz"**.**toCharArray**();**

StringBuilder sb **=** **new** StringBuilder**();**

**for** **(**int nu **=** 0**;** nu **<** 6**;** nu**++)** **{**

char c **=** chars**[**rand**.**nextInt**(**chars**.**length**)];**

sb**.**append**(**c**);**

**}**

// transactie

**try** **{**

Statement st **=** **(**Statement**)** connection**.**createStatement**();**

int studentnummer **=** 1876814 **+** i**;**

int modulenummer **=** i **+** 10**;**

String klas **=** "INF" **+** i **+** "D"**;**

String naam **=** sb**.**toString**();**

String modulenaam **=** "INFANL-" **+** modulenummer**;**

st**.**executeUpdate**(**"INSERT INTO Student "

**+** "VALUES ('"

**+** studentnummer

**+** "','"

**+** naam

**+** "' ,'Lam' ,'' , '1996-02-28' ,'man' ,'wijnhaven',106 ,'','Rotterdam','2222AA','0639387945')"**);**

int n **=** rand**.**nextInt**(**30**)** **+** 1**;**

**if** **(**n **==** 20**)** **{**

groep**.**add**(**klas**);**

st**.**executeUpdate**(**"INSERT INTO Groep " **+** "VALUES ('"

**+** klas **+** "', '2015-05-05', '2016-01-01')"**);**

**}**

// student aan willekeurige klas

st**.**executeUpdate**(**"INSERT INTO Groep\_has\_Student "

**+** "VALUES ('"

**+** groep**.**get**(**rand**.**nextInt**(**groep**.**size**()))**

**+** "' ,'" **+** studentnummer **+** "')"**);**

int z **=** rand**.**nextInt**(**30**)** **+** 1**;**

**if** **(**z **==** 20**)** **{**

module**.**add**(**modulenaam**);**

st**.**executeUpdate**(**"INSERT INTO Cursussen "

**+** "VALUES ('"

**+** modulenaam

**+** "', 'TJANG','analyse','2015-05-05', '2016-01-01')"**);**

int a **=** rand**.**nextInt**(**100**)** **+** 1**;**

**if** **(**a **<=** 15**)** **{**

**for** **(**int b **=** 0**;** b **<** groep**.**size**();** b**++)** **{**

st**.**executeUpdate**(**"INSERT INTO Cursussen\_has\_Groep "

**+** "VALUES ('"

**+** groep**.**get**(**b**)**

**+** "','" **+** modulenaam **+** "')"**);**

**}**

**}**

**}**

**}** **catch** **(**SQLException e1**)** **{**

// TODO Auto-generated catch block

e1**.**printStackTrace**();**

**}**

// sla eind tijd op

eindtijd **=** System**.**currentTimeMillis**();**

// Bereken duur

duurInms **=** eindtijd **-** begintijd**;**

System**.**out**.**println**(**"Gebruiker 1 : " **+** duurInms**);**

totale\_duur\_thread1**.**add**(**duurInms**);**

**}**

int i**;**

float totale\_duur **=** 0**;**

**for** **(**i **=** 1**;** i **<** totale\_duur\_thread1**.**size**();** i**++)** **{**

totale\_duur **+=** totale\_duur\_thread1**.**get**(**i**);**

**}**

System**.**out**.**println**(**"thread 1 totale duur : " **+** totale\_duur**);**

System**.**out**.**println**(**"thread 1 gemiddeld: " **+** **(**totale\_duur **/** totale\_duur\_thread1**.**size**()));**

**try** **{**

connection**.**commit**();**

connection**.**close**();**

**}** **catch** **(**SQLException e**)** **{**

// TODO Auto-generated catch block

e**.**printStackTrace**();**

**}**

**try** **{**

Thread**.**sleep**((**long**)** **(**Math**.**random**()** **\*** 1000**));**

**}** **catch** **(**InterruptedException e**)** **{**

// TODO Auto-generated catch block

e**.**printStackTrace**();**

**}**

**}**

**},** "Thread 1"**).**start**();**

// Maak en start thread 2

new Thread(new Runnable() {

@Override

public void run() {

Connection connection = connect();

// Decladeer Thread 2 variables

List<String> groep = new ArrayList<String>();

List<String> module = new ArrayList<String>();

List<Long> totale\_duur\_thread2 = new ArrayList<Long>();

// Decladeer variables

long begintijd;

long eindtijd;

long duurInms;

// Onze eerste klas en module

Statement st1;

try {

st1 = (Statement) connection.createStatement();

st1.executeUpdate("INSERT INTO Groep "

+ "VALUES ('INF1B', '2015-05-05', '2016-01-01')");

groep.add("INF1B");

st1.executeUpdate("INSERT INTO Cursussen "

+ "VALUES ('INFDEV-01','TJANG','analyse','2015-05-05', '2016-01-01')");

module.add("INFDEV-01");

} catch (SQLException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

// Begin 600 iteraties van student en klas toevoegen

for (int i = 0; i < 600; i++) {

// sla begin tijd op

begintijd = System.currentTimeMillis();

// random string

char[] chars = "abcdefghijklmnopqrstuvwxyz".toCharArray();

StringBuilder sb = new StringBuilder();

for (int nu = 0; nu < 6; nu++) {

char c = chars[rand.nextInt(chars.length)];

sb.append(c);

}

// transactie

try {

Statement st = (Statement) connection.createStatement();

int studentnummer = 2876814 + i;

int modulenummer = i + 10;

String klas = "INF" + i + "B";

String naam = sb.toString();

String modulenaam = "INFDEV-" + modulenummer;

st.executeUpdate("INSERT INTO Student "

+ "VALUES ('"

+ studentnummer

+ "','"

+ naam

+ "' ,'Lam' ,'' , '1996-02-28' ,'man' ,'wijnhaven',106 ,'','Rotterdam','2222AA','0639387945')");

int n = rand.nextInt(30) + 1;

if (n == 20) {

groep.add(klas);

st.executeUpdate("INSERT INTO Groep " + "VALUES ('"

+ klas + "', '2015-05-05', '2016-01-01')");

}

// student aan willekeurige klas

st.executeUpdate("INSERT INTO Groep\_has\_Student "

+ "VALUES ('"

+ groep.get(rand.nextInt(groep.size()))

+ "' ,'" + studentnummer + "')");

int z = rand.nextInt(30) + 1;

if (z == 20) {

module.add(modulenaam);

st.executeUpdate("INSERT INTO Cursussen "

+ "VALUES ('"

+ modulenaam

+ "', 'TJANG','analyse','2015-05-05', '2016-01-01')");

int a = rand.nextInt(100) + 1;

if (a <= 15) {

for (int b = 0; b < groep.size(); b++) {

st.executeUpdate("INSERT INTO Cursussen\_has\_Groep "

+ "VALUES ('"

+ groep.get(b)

+ "','" + modulenaam + "')");

}

}

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

// sla eind tijd op

eindtijd = System.currentTimeMillis();

// Bereken duur

duurInms = eindtijd - begintijd;

System.out.println("Gebruiker 2 : " + duurInms);

totale\_duur\_thread2.add(duurInms);

}

int i;

float totale\_duur = 0;

for (i = 1; i < totale\_duur\_thread2.size(); i++) {

totale\_duur += totale\_duur\_thread2.get(i);

}

System.out.println("thread 2 totale duur : " + totale\_duur);

System.out.println("thread 2 gemiddeld: " + (totale\_duur / totale\_duur\_thread2.size()));

try {

connection.commit();

connection.close();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

try {

Thread.sleep((long) (Math.random() \* 1000));

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}, "Thread 2").start();

// Maak en start thread 3

new Thread(new Runnable() {

@Override

public void run() {

Connection connection = connect();

List<String> groep = new ArrayList<String>();

List<String> module = new ArrayList<String>();

// Decladeer variables

long begintijd;

long eindtijd;

long duurInms;

List<Long> totale\_duur\_thread3 = new ArrayList<Long>();

// Onze eerste klas en module

Statement st1;

try {

st1 = (Statement) connection.createStatement();

st1.executeUpdate("INSERT INTO Groep "

+ "VALUES ('INF1C', '2015-05-05', '2016-01-01')");

groep.add("INF1C");

st1.executeUpdate("INSERT INTO Cursussen "

+ "VALUES ('INFONZ-01','TJANG','analyse','2015-05-05', '2016-01-01')");

module.add("INFONZ-01");

} catch (SQLException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

// Begin 600 iteraties van student en klas toevoegen

for (int i = 0; i < 600; i++) {

// sla begin tijd op

begintijd = System.currentTimeMillis();

// random string

char[] chars = "abcdefghijklmnopqrstuvwxyz".toCharArray();

StringBuilder sb = new StringBuilder();

for (int nu = 0; nu < 6; nu++) {

char c = chars[rand.nextInt(chars.length)];

sb.append(c);

}

// transactie

try {

Statement st = (Statement) connection.createStatement();

int studentnummer = 3876814 + i;

int modulenummer = i + 10;

String klas = "INF" + i + "C";

String naam = sb.toString();

String modulenaam = "INFONZ-" + modulenummer;

st.executeUpdate("INSERT INTO Student "

+ "VALUES ('"

+ studentnummer

+ "','"

+ naam

+ "' ,'Lam' ,'' , '1996-02-28' ,'man' ,'wijnhaven',106 ,'','Rotterdam','2222AA','0639387945')");

int n = rand.nextInt(30) + 1;

if (n == 20) {

groep.add(klas);

st.executeUpdate("INSERT INTO Groep " + "VALUES ('"

+ klas + "', '2015-05-05', '2016-01-01')");

}

// student aan willekeurige klas

st.executeUpdate("INSERT INTO Groep\_has\_Student "

+ "VALUES ('"

+ groep.get(rand.nextInt(groep.size()))

+ "' ,'" + studentnummer + "')");

int z = rand.nextInt(30) + 1;

if (z == 20) {

module.add(modulenaam);

st.executeUpdate("INSERT INTO Cursussen "

+ "VALUES ('"

+ modulenaam

+ "', 'TJANG','analyse','2015-05-05', '2016-01-01')");

int a = rand.nextInt(100) + 1;

if (a <= 15) {

for (int b = 0; b < groep.size(); b++) {

st.executeUpdate("INSERT INTO Cursussen\_has\_Groep "

+ "VALUES ('"

+ groep.get(b)

+ "','" + modulenaam + "')");

}

}

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

// sla eind tijd op

eindtijd = System.currentTimeMillis();

// Bereken duur

duurInms = eindtijd - begintijd;

System.out.println("Gebruiker 3 : " + duurInms);

totale\_duur\_thread3.add(duurInms);

}

int i;

float totale\_duur = 0;

for (i = 1; i < totale\_duur\_thread3.size(); i++) {

totale\_duur += totale\_duur\_thread3.get(i);

}

System.out.println("thread 3 totale duur : " + totale\_duur);

System.out.println("thread 3 gemiddeld: " + (totale\_duur / totale\_duur\_thread3.size()));

try {

connection.commit();

connection.close();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

try {

Thread.sleep((long) (Math.random() \* 1000));

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}, "Thread 3").start();

}

//Connect functie

public static Connection connect() {

// maak verbinding met postgres

// maak verbinding met de driver

try {

Class.forName("org.postgresql.Driver");

} catch (ClassNotFoundException e) {

throw new RuntimeException(

"Cannot find the driver in the classpath!", e);

}

// maak verbinding met de database

Connection connection = null;

try {

System.out.println("Connecting database...");

connection = DriverManager.getConnection(url, username, password);

System.out.println("Database connected!");

connection.setAutoCommit(false);

} catch (Exception e) {

}

return connection;

}

}

# Simulatie: gebruikers die data opvragen

Hieronder vindt U de JAVA code die we hebben geschreven om gelijktijdig meerdere data op te vragen uit de POSTGRES database. Er wordt per iteratie geprint hoelang elke thread erover doet.

We hebben in elke thread een aantal SELECTS uitgevoerd. De gemeten tijd per iteratie ligt tussen de 0 en 4 milliseconden.

Uiteindelijk wordt de tijd per iteratie uitgeprint samen met de vakken die een random gekozen student volgt / heeft gevolg. Aan het eind van elke thread wordt de gemiddelde tijd van de iteraties uitgeprint.

**import** java**.**sql**.**Connection**;**

**import** java**.**sql**.**DriverManager**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**sql**.**Statement**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**Random**;**

public class DataLezen **{**

public static void main**(**String**[]** args**)** **{**

// Maak en start thread 1

**new** Thread**(new** Runnable**()** **{**

@Override

public void run**()** **{**

Connection connection **=** DataInvoegen**.**connect**();**

// Decladeer variables

long begintijd**;**

long eindtijd**;**

long duurInms**;**

List**<**Long**>** totale\_duur\_thread **=** **new** ArrayList**<**Long**>();**

Statement st1**;**

// Begin 600 iteraties van student en klas toevoegen

**for** **(**int i **=** 0**;** i **<** 600**;** i**++)** **{**

// SELECT ALLE STUDENTEN

**try** **{**

st1 **=** **(**Statement**)** connection**.**createStatement**();**

ResultSet z**;**

z **=** st1**.**executeQuery**(**"SELECT voornaam,achternaam FROM Student ORDER BY RANDOM() LIMIT 1;"**);**

String voornaam **=** **null;**

String achternaam **=** **null;**

String groepnaam **=** **null;**

String cursusnaam **=** **null;**

**if** **(**z**.**next**())** **{**

voornaam **=** z**.**getString**(**"voornaam"**);**

achternaam **=** z**.**getString**(**"achternaam"**);**

**}**

// sla begin tijd op

begintijd **=** System**.**currentTimeMillis**();**

// transactie

ResultSet groep\_ophalen**;**

groep\_ophalen **=** st1

**.**executeQuery**(**"SELECT groep\_groepnaam FROM Student INNER JOIN Groep\_has\_Student ON Student.studentnummer = Groep\_has\_Student.student\_studentnummer WHERE voornaam = '"

**+** voornaam

**+** "' AND achternaam = '"

**+** achternaam **+** "';"**);**

**if** **(**groep\_ophalen**.**next**())** **{**

groepnaam **=** groep\_ophalen

**.**getString**(**"groep\_groepnaam"**);**

**}**

ResultSet cursus\_ophalen**;**

cursus\_ophalen **=** st1

**.**executeQuery**(**"SELECT \* FROM Cursussen\_has\_Groep WHERE groep\_groepnaam = '"

**+** groepnaam **+** "';"**);**

**while** **(**cursus\_ophalen**.**next**())** **{**

cursusnaam **=** cursus\_ophalen

**.**getString**(**"cursussen\_cursuscode"**);**

System**.**out**.**println**(**"A: " **+** voornaam **+** " "

**+** achternaam **+** " in " **+** groepnaam

**+** " volgt " **+** cursusnaam**);**

**}**

// sla eind tijd op

eindtijd **=** System**.**currentTimeMillis**();**

// Bereken duur

duurInms **=** eindtijd **-** begintijd**;**

System**.**out**.**println**(**"Gebruiker 1 : " **+** duurInms**);**

totale\_duur\_thread**.**add**(**duurInms**);**

**}** **catch** **(**SQLException e**)** **{**

// TODO Auto-generated catch block

e**.**printStackTrace**();**

**}**

**}**

int i**;**

float totale\_duur **=** 0**;**

**for** **(**i **=** 1**;** i **<** totale\_duur\_thread**.**size**();** i**++)** **{**

totale\_duur **+=** totale\_duur\_thread**.**get**(**i**);**

**}**

System**.**out**.**println**(**"thread 1 deed : " **+** totale\_duur**);**

System**.**out**.**println**(**"thread 1 gemiddeld: " **+** **(**totale\_duur **/** totale\_duur\_thread**.**size**()));**

**try** **{**

connection**.**commit**();**

connection**.**close**();**

**}** **catch** **(**SQLException e**)** **{**

// TODO Auto-generated catch block

e**.**printStackTrace**();**

**}**

**}**

**},** "Thread 1"**).**start**();**

// Maak en start thread 2

**new** Thread**(new** Runnable**()** **{**

@Override

public void run**()** **{**

Connection connection **=** DataInvoegen**.**connect**();**

// Decladeer variables

long begintijd**;**

long eindtijd**;**

long duurInms**;**

List**<**Long**>** totale\_duur\_thread **=** **new** ArrayList**<**Long**>();**

Statement st1**;**

// Begin 600 iteraties van student en klas toevoegen

**for** **(**int i **=** 0**;** i **<** 600**;** i**++)** **{**

// SELECT ALLE STUDENTEN

**try** **{**

st1 **=** **(**Statement**)** connection**.**createStatement**();**

ResultSet z**;**

z **=** st1**.**executeQuery**(**"SELECT voornaam,achternaam FROM Student ORDER BY RANDOM() LIMIT 1;"**);**

String voornaam **=** **null;**

String achternaam **=** **null;**

String groepnaam **=** **null;**

String cursusnaam **=** **null;**

**if** **(**z**.**next**())** **{**

voornaam **=** z**.**getString**(**"voornaam"**);**

achternaam **=** z**.**getString**(**"achternaam"**);**

**}**

// sla begin tijd op

begintijd **=** System**.**currentTimeMillis**();**

// transactie

ResultSet groep\_ophalen**;**

groep\_ophalen **=** st1

**.**executeQuery**(**"SELECT groep\_groepnaam FROM Student INNER JOIN Groep\_has\_Student ON Student.studentnummer = Groep\_has\_Student.student\_studentnummer WHERE voornaam = '"

**+** voornaam

**+** "' AND achternaam = '"

**+** achternaam **+** "';"**);**

**if** **(**groep\_ophalen**.**next**())** **{**

groepnaam **=** groep\_ophalen

**.**getString**(**"groep\_groepnaam"**);**

**}**

ResultSet cursus\_ophalen**;**

cursus\_ophalen **=** st1

.executeQuery("SELECT \* FROM Cursussen\_has\_Groep WHERE groep\_groepnaam = '"

+ groepnaam + "';");

while (cursus\_ophalen.next()) {

cursusnaam = cursus\_ophalen

.getString("cursussen\_cursuscode");

System.out.println("B: " + voornaam + " "

+ achternaam + " in " + groepnaam

+ " volgt " + cursusnaam);

}

// sla eind tijd op

eindtijd = System.currentTimeMillis();

// Bereken duur

duurInms = eindtijd - begintijd;

System.out.println("Gebruiker 2 : " + duurInms);

totale\_duur\_thread.add(duurInms);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

int i;

float totale\_duur = 0;

for (i = 1; i < totale\_duur\_thread.size(); i++) {

totale\_duur += totale\_duur\_thread.get(i);

}

System.out.println("thread 2 totale duur : " + totale\_duur);

System.out.println("thread 2 gemiddeld: " + (totale\_duur / totale\_duur\_thread.size()));

try {

connection.commit();

connection.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}, "Thread 2").start();

}

}

# Simulatie: Indexen

Hieronder vindt U de JAVA code die we hebben geschreven om aan te tonen dat het gebruiken van een index een positieve performance impact heeft. We hebben allereerst de INDEX verwijdert. Hierna hebben we alle studenten opgevraagd van wie de voornaam begint met een “a”. de tijd voor het ophalen is gemeten. Hierna hebben we een INDEX gemaakt voor de “voornaam” kolom in de “student” tabel. Hierna hebben we nogmaals alle studenten opgevraagd van wie de voornaam begint met een “a”. Ook hier werd de tijd gemeten. In de resultaten onderaan de code ziet U dat de INDEX wordt gebruikt en dat het een positieve impact heeft op de performance.

**import** java**.**sql**.**Connection**;**

**import** java**.**sql**.**DriverManager**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**sql**.**Statement**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**Random**;**

public class DataLezenIndex **{**

public static void main**(**String**[]** args**)** **{**

// Maak en start thread 1

**new** Thread**(new** Runnable**()** **{**

@Override

public void run**()** **{**

Connection connection **=** DataInvoegen**.**connect**();**

// Decladeer variables

long begintijd\_zonder\_idx**;**

long eindtijd\_zonder\_idx**;**

long begintijd\_met\_idx**;**

long eindtijd\_met\_idx**;**

long duurInms\_zonder\_idx**;**

long duurInms\_met\_idx**;**

Statement st1**;**

String voornaam **=** **null;**

**try** **{**

st1 **=** **(**Statement**)** connection**.**createStatement**();**

//Zonder IDX duur

st1**.**executeUpdate**(**"DROP INDEX Student\_idx;"**);**

// sla begin tijd op

begintijd\_zonder\_idx **=** System**.**currentTimeMillis**();**

ResultSet z**;**

z **=** st1**.**executeQuery**(**"SELECT voornaam FROM Student WHERE voornaam LIKE 'a%';"**);**

**if** **(**z**.**next**())** **{**

voornaam **=** z**.**getString**(**"voornaam"**);**

**}**

eindtijd\_zonder\_idx **=** System**.**currentTimeMillis**();**

duurInms\_zonder\_idx **=** eindtijd\_zonder\_idx **-** begintijd\_zonder\_idx**;**

System**.**out**.**println**(**"Zonder Index Duurt : " **+** duurInms\_zonder\_idx**);**

//Met IDX duur

st1**.**executeUpdate**(**"CREATE UNIQUE INDEX Student\_idx ON Student (voornaam);"**);**

// sla begin tijd op

begintijd\_met\_idx **=** System**.**currentTimeMillis**();**

ResultSet b**;**

b **=** st1**.**executeQuery**(**"SELECT voornaam FROM Student WHERE voornaam LIKE 'a%';"**);**

**if** **(**b**.**next**())** **{**

voornaam **=** b**.**getString**(**"voornaam"**);**

**}**

eindtijd\_met\_idx **=** System**.**currentTimeMillis**();**

duurInms\_met\_idx **=** eindtijd\_met\_idx **-** begintijd\_met\_idx**;**

System**.**out**.**println**(**"Met Index Duurt : " **+** duurInms\_met\_idx**);**

connection**.**close**();**

**}** **catch** **(**SQLException e**)** **{**

// TODO Auto-generated catch block

e**.**printStackTrace**();**

**}**

**}**

**},** "Thread 1"**).**start**();**

**}**

**}**

Hier ziet U dat de cost veel lager is wanneer we een INDEX gebruiken.

