

## **Overview**

In this third CS1003 practical, I was asked to complete 6 requirements:

- Create an ER diagram
- Create a Relational Schema
- Create a Java program to initialise the Relational Database
- Create a Java program to populate the database
- Create a Java program to query the database
- Write the project report

I also had to test my code and made sure that it works. Some tests are implemented in Java, others are through DBeaver.

I'm proud to say that I've achieved the previously established goals:

- I have completed all 6 requirements
- All tests are successful (see **Testing** for more details)

## **Design**

The project is made up of an SQLite3 database, database.db, its ER Chen diagram, a Data Definition Language description for the database (DDL.txt), and four Java programs (InitialiseDB.java, PopulateDB.java, Query.java and Test.java) in a src folder. Each file is for a specific requirement. I will now describe each file, except test.java, which shall be explained in the **Testing** part of this report, and database.db.

The ER Diagram is made of four entities: Persons, Links, Publications and Venues. As one can guess, the Persons table stores authors, Publications for publications and Venues for venues. As one author can have multiple publications, but one publication can have multiple authors, I had to include a many-to-many relationship, Links, to link the two entities. Each Person has a primary key, its ID (pid), and a name. Each Publication has a primary key, its key (pubkey), a title (title), a year of publication (pubyear), and a foreign key, its venue's key (venkey). Each Venue has a primary key, its key (venkey), and a title (title). Finally, each Link has two foreign keys: the author's id (pid) and the publication's key (pubkey). See ER\_Chen.jpg for the ER Diagram (made with draw.io).

To create the Relational Schema in the Requirement 2, I created the database modeled in the ER Diagram using DBeaver. The "Generate SQL > DDL" tool helped me generate the DDL file for the database. See DDL.txt for the database's Data Definition Language description.

The first Java file, InitialiseDB.java, initialises the database by creating the four tables. This can be divided into four steps: firstly, the program checks whether the database already exists and deletes it if so. Secondly, it creates a new SQLite database. Thirdly, it creates the tables by reading the DDL.txt file line by line using a Scanner object and JDBC's *executeUpdate* method. Finally, it checks if those four tables have been initialised properly with the *checkInitial* method, that calls queries the sqlite\_master for the name of all initialised tables: if the query returns the names of all four tables, then the method returns "OK", otherwise, it returns "Not OK".

The Requirement 4 is covered in the second Java file, PopulateDB.java. It has a private Connection attribute, *connection*, to connect to the database. This class re-uses numerous methods from the previous CS1003 Practical to check the cache or call the DBLP database. The

*getURL*, *getEncodedURL*, *getSavedResponse* and *saveResponse* methods are identical to those in the CS1003 Practical 2. The format of the *searchAuthor*, *searchPublication* and *searchVenue* methods is very similar to the format of the *getVenue* method from the previous practical: they search the DBLP database (or the files in the cache folder) and get the information required. However, instead of printing out the information found from the xml files to the terminal, the code adds these values to the database using *insertAuthor*, *insertPublication*, *insertLink* and *insertVenue*.

The methods are nested in each other: this is to make sure we have the all the information required to insert values in the database. For every *searchAuthor* call, we insert the values retrieved in the Persons table, and we call the *searchPublication* method. For every *searchPublication* call, we first call the *searchVenue* method, then we insert the values retrieved in the Publications, and finally, we insert the values retrieved in the Links table. For every *searchVenue* call, we simply insert the values retrieved in the Venues table. The order here is important, as we need to insert the values in tables with primary keys before those with secondary keys (i.e., filling the Venues table before the Publications table, and filling the Authors and Publications tables before the Links table).

For this class to work, the *InitialiseDB.java* file must be run before.

The final requirement, Requirement 5, is to query the database. This is done in the *QueryDB.java* file. It has 5 methods, each for one query. They all take a *Connection* object as parameter to connect to the database. The user inputs the query number as command line argument: to access this number, the program uses a *switch... case... default...* in the *main* method. If the user inputs an invalid query number, then a custom error message is returned. To query the database, the program uses JDBC's *Statement* objects and *executeQuery* method, and prints out the results to the terminal.

For this class to work, the *InitialiseDB.java* and *PopulateDB.java* must be run before.

## **Testing**

To implement additional tests, I created a separate Java file called *Test.java*. This file contains four methods, each to test a different aspect of my submission. They all take a *Connection* object as parameter to connect to the database. The first method is to test if the four tables have been initialised properly. The second method is to test if all four authors have been added to the Persons table. The third method is to test if the two publications given as example, "Investigating Binary Partition Power in Metric Query." and "Bitpart: Exact metric search in high(er) dimensions.", have been added to the Publications table. The fourth and final method is to test if the two venues given as example, "30<sup>th</sup> SEBD, 2022: Tirrenia, Italy" and "Information Systems, Volume 95 January 2021", have been added to the Venues table. The output is as follows:

Test if the four tables have been initialised: OK

Test if the four authors have been added to the "Persons" table: OK

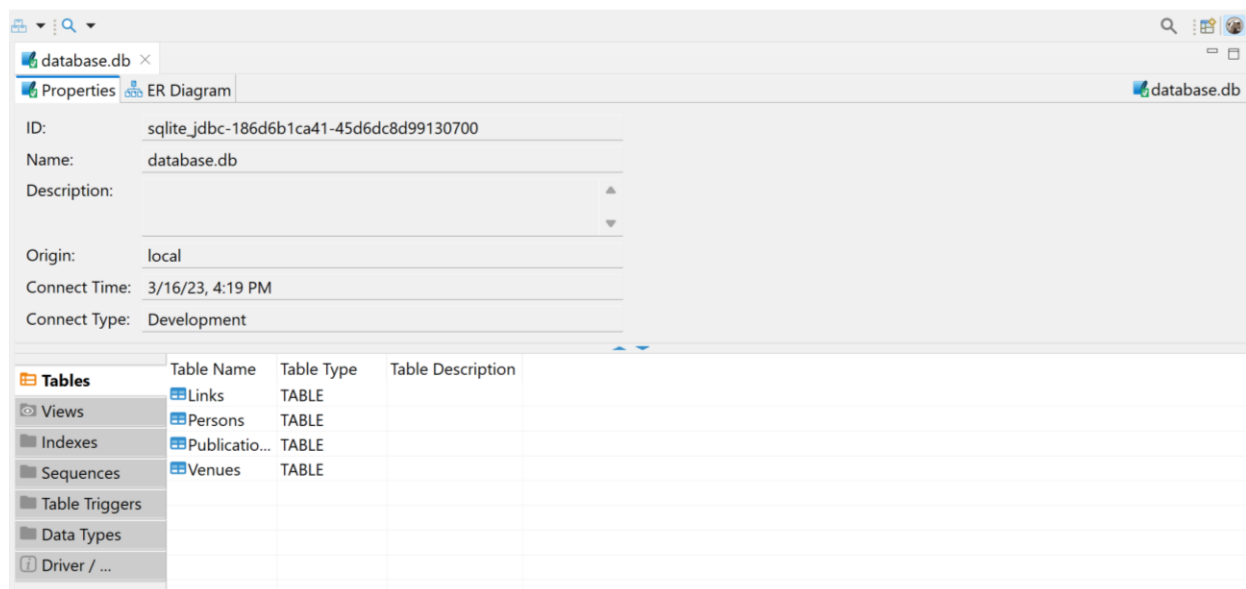
Test if "Investigating Binary Partition Power in Metric Query." and "Bitpart: Exact metric search in high(er) dimensions.", have been added to the "Publications" table: OK

Test if "30<sup>th</sup> SEBD, 2022: Tirrenia, Italy" and "Information Systems, Volume 95 January 2021", have been added to the "Venues" table: OK

The following table illustrates other tests that I implemented:

What is being tested	Name of test method	Pre conditions	Expected outcome	Actual outcome	Evidence
Database creation and initialisation	None	The database doesn't exist	The database is created and initialised	The database is created, as well as the tables	Use ls command for the creation  See Figure 2 for the initialisation
Database population: Publications	None	The database exists and has been initialised	There are 256 distinct publications in the Publications table	The Publications table has been initialised properly	See Figure 3
Database population: Links	None	The database exists and has been initialised	There are 268 links in the Links table	The Links table has been initialised properly	See Figure 4
Database population: Venues	None	The database exists and has been initialised	There are 190 distinct venues in the Venues table	The Venues table has been initialised properly	See Figure 5

Figure 1: Testing table



The screenshot shows a database management interface. The top section displays connection properties for 'database.db':

- ID: sqlite\_jdbc-186d6b1ca41-45d6dc8d99130700
- Name: database.db
- Description: (empty)
- Origin: local
- Connect Time: 3/16/23, 4:19 PM
- Connect Type: Development

The bottom section shows a list of tables:

Table Name	Table Type	Table Description
Links	TABLE	
Persons	TABLE	
Publicatio...	TABLE	
Venues	TABLE	

Figure 2: The tables have been initialised

pubkey	pubyear	title	venkey
conf/cpaior/AkgunDMSS20	2,020	Discriminating Instance Generation from Abstract	db/conf/cpaior/cpaior2020
conf/ecai/KocakAGM20	2,020	Exploiting Incomparability in Solution Dominance	db/conf/ecai/ecai2020
conf/cp/AkgunDMSS19	2,019	Instance Generation via Generator Instances	db/conf/cp/cp2019
conf/cp/SpracklenDAM19	2,019	Automatic Streamlining for Constrained Optimisat	db/conf/cp/cp2019
conf/cp/SpracklenAM18	2,018	Automatic Generation and Selection of Streamline	db/conf/cp/cp2018
conf/icdm/KocakAMN18	2,018	Closed Frequent Itemset Mining with Arbitrary Sid	db/conf/icdm/icdm2018w
conf/cp/WetterAM15	2,015	Automatically Generating Streamlined Constraint	db/conf/cp/cp2015
conf/eScience/WetterABDMV15	2,015	Cloud-based E-Infrastructure for Scheduling Astro	db/conf/eScience/eScience2015
conf/cloudcom/VargheseAMTB14	2,014	Cloud Benchmarking for Performance	db/conf/cloudcom/cloudcom2014
conf/cloudcom/ThaiBVAM14	2,014	Optimal Deployment of Geographically Distribute	db/conf/cloudcom/cloudcom2014
conf/aaai/AkgunMJFH11	2,011	Extensible Automated Constraint Modelling	db/conf/aaai/aaai2011
journals/moc/AkgunMV22	2,022	Enumeration of set-theoretic solutions to the Yang	db/journals/moc/moc91
journals/corr/abs-2205-14753	2,022	A Framework for Generating Informative Benchma	db/journals/corr/corr2205
journals/tvcg/ZhuNAN20	2,020	How People Visually Represent Discrete Constrai	db/journals/tvcg/tvcg26
journals/corr/abs-2009-10152	2,020	Towards Portfolios of Streamlined Constraint Mod	db/journals/corr/corr2009
journals/corr/abs-2009-10156	2,020	Exploring Instance Generation for Automated Plan	db/journals/corr/corr2009
journals/corr/abs-2009-11111	2,020	Efficient Incremental Modelling and Solving	db/journals/corr/corr2009
journals/tcc/VargheseAMTB19	2,019	Cloud Benchmarking for Maximising Performance	db/journals/tcc/tcc7
journals/corr/abs-1910-00475	2,019	Conjure Documentation, Release 2.3.0	db/journals/corr/corr1910
journals/corr/abs-1910-00505	2,019	Towards Improving Solution Dominance with Inco	db/journals/corr/corr1910
journals/corr/abs-1808-09847	2,018	Modelling Langford's Problem: A Viewpoint for Se	db/journals/corr/corr1808
journals/corr/abs-1808-09870	2,018	Memory Consistency Models using Constraints	db/journals/corr/corr1808
journals/constraints/Akgun17	2,017	Extensible automated constraint modelling via refi	db/journals/constraints/constraints22
journals/corr/abs-1708-01829	2,017	Declarative Statistics	db/journals/corr/corr1708
journals/corr/VargheseAMTB16	2,016	Cloud Benchmarking For Maximising Performance	db/journals/corr/corr1608
journals/corr/RossiAPT16	2,016	The BIN_COUNTS Constraint: Filtering and Applica	db/journals/corr/corr1611
journals/corr/ThaiBVAM14	2,014	Optimal Deployment of Geographically Distribute	db/journals/corr/corr1410
journals/corr/VargheseAMTB14	2,014	Cloud Benchmarking for Performance	db/journals/corr/corr1411
journals/corr/abs-1109-1774	2,011	Conjure Revisited: Towards Automated Constraint	db/journals/corr/corr1109

Figure 3: The last 29 entries of the Publications table (it has 256 entries)

pid	pubkey	title
35/9923	conf/cloudcom/VargheseAMTB14	
35/9923	conf/cloudcom/ThaiBVAM14	
35/9923	conf/cp/NightingaleAGJM14	
35/9923	conf/ecai/AkgunGJMN14	
35/9923	conf/cp/AkgunFGHJMN13	
35/9923	conf/aaai/AkgunMJFH11	
35/9923	journals/ai/AkgunFGJMN22	
35/9923	journals/moc/AkgunMV22	
35/9923	journals/corr/abs-2202-13250	
35/9923	journals/corr/abs-2205-14753	
35/9923	journals/corr/abs-2111-00821	
35/9923	journals/tvcg/ZhuNAN20	
35/9923	journals/corr/abs-2009-10152	
35/9923	journals/corr/abs-2009-10156	
35/9923	journals/corr/abs-2009-11111	
35/9923	journals/jintseq/AkgunGKZ19	
35/9923	journals/tcc/VargheseAMTB19	
35/9923	journals/corr/abs-1910-00475	
35/9923	journals/corr/abs-1910-00505	
35/9923	journals/corr/abs-1808-09847	
35/9923	journals/corr/abs-1808-09870	
35/9923	journals/ai/NightingaleAGJM17	
35/9923	journals/constraints/Akgun17	
35/9923	journals/corr/abs-1708-01829	
35/9923	journals/corr/VargheseAMTB16	
35/9923	journals/corr/RossiAPT16	
35/9923	journals/corr/ThaiBVAM14	
35/9923	journals/corr/VargheseAMTB14	
35/9923	journals/corr/abs-1109-1774	

Figure 4: The last 29 entries of the Links table (it has 269 entries)

venuekey	title
db/journals/sLogica/sLogica52	Studia Logica, Volume 52
db/journals/corr/corr9309	CoRR, September 1993
db/journals/ndjfl/ndjfl33	Notre Dame Journal of Formal Logic, Volume 33
db/conf/ipres/ipres2015	iPRES 2015: Chapel Hill, North Carolina, USA
db/conf/cp/cp2022	28th CP 2022: Haifa, Israel
db/conf/cpaior/cpaior2021	18th CPAIOR 2021: Vienna, Austria
db/conf/cp/cp2020	26th CP 2020: Louvain-la-Neuve, Belgium
db/conf/cpaior/cpaior2020	17th CPAIOR 2020: Vienna, Austria
db/conf/ecai/ecai2020	24th ECAI 2020: Santiago de Compostela, Spain
db/conf/cp/cp2019	25th CP 2019: Stamford, CT, USA
db/conf/icdm/icdm2018w	18th ICDM 2018: Singapore - Workshops
db/conf/cp/cp2015	21. CP 2015: Cork, Ireland
db/conf/eScience/eScience2015	11th eScience 2015: Munich, Germany
db/conf/cloudcom/cloudcom2014	CloudCom 2014: Singapore
db/conf/aaai/aaai2011	25th AAAI 2011: San Francisco, California, USA
db/journals/moc/moc91	Mathematics of Computation, Volume 91
db/journals/corr/corr2205	CoRR, May 2022
db/journals/tvcg/tvcg26	IEEE Transactions on Visualization and Computer Graphics, Volume 26
db/journals/corr/corr2009	CoRR, September 2020
db/journals/tcc/tcc7	IEEE Transactions on Cloud Computing, Volume 7
db/journals/corr/corr1910	CoRR, October 2019
db/journals/corr/corr1808	CoRR, August 2018
db/journals/constraints/constraints22	Constraints - An International Journal, Volume 22
db/journals/corr/corr1708	CoRR, August 2017
db/journals/corr/corr1608	CoRR, August 2016
db/journals/corr/corr1611	CoRR, November 2016
db/journals/corr/corr1410	CoRR, October 2014
db/journals/corr/corr1411	CoRR, November 2014
db/journals/corr/corr1109	CoRR, September 2011

Figure 5: The last 29 entries of the Venues table (it has 190 entries)

## Examples

Here are the outputs of the five queries in the QueryDB.java file:

```
>> java InitialiseDB "../database.db"
```

OK

```
>> java PopulateDB "../database.db"
```

```
>> java QueryDB 1
```

Total number of publications that have been published by either "Özgür Akgün" or "Ian Gent" or "Alan Dearle": 256

```
>> java QueryDB 2
```

Total number of publications that have been published by "Özgür Akgün": 46

```
>> java QueryDB 3
```

Titles of the publications of "Özgür Akgün":

A Framework for Generating Informative Benchmark Instances.

Understanding How People Approach Constraint Modelling and Solving.

Finding Subgraphs with Side Constraints.

Effective Encodings of Constraint Programming Models to SMT.

Discriminating Instance Generation from Abstract Specifications: A Case Study with CP and MIP.

Exploiting Incomparability in Solution Dominance: Improving General Purpose Constraint-Based Mining.

Instance Generation via Generator Instances.

Automatic Streamlining for Constrained Optimisation.

Automatic Discovery and Exploitation of Promising Subproblems for Tabulation.



Automatic Generation and Selection of Streamlined Constraint Models via Monte Carlo Search on a Model Lattice.

Metamorphic Testing of Constraint Solvers.

Closed Frequent Itemset Mining with Arbitrary Side Constraints.

A Framework for Constraint Based Local Search using Essence.

Using Metric Space Indexing for Complete and Efficient Record Linkage.

Exploiting Short Supports for Improved Encoding of Arbitrary Constraints into SAT.

Automatically Generating Streamlined Constraint Models with Essence and Conjure.

Cloud-based E-Infrastructure for Scheduling Astronomical Observations.

Cloud Benchmarking for Performance.

Optimal Deployment of Geographically Distributed Workflow Engines on the Cloud.

Automatically Improving Constraint Models in Savile Row through Associative-Commutative Common Subexpression Elimination.

Breaking Conditional Symmetry in Automated Constraint Modelling with CONJURE.

Automated Symmetry Breaking and Model Selection in Conjure.

Extensible Automated Constraint Modelling.

Conjure: Automatic Generation of Constraint Models from Problem Specifications.

Enumeration of set-theoretic solutions to the Yang-Baxter equation.

Automatic Tabulation in Constraint Models.

A Framework for Generating Informative Benchmark Instances.

Towards Reformulating Essence Specifications for Robustness.

How People Visually Represent Discrete Constraint Problems.

Towards Portfolios of Streamlined Constraint Models: A Case Study with the Balanced Academic Curriculum Problem.

Exploring Instance Generation for Automated Planning.

Efficient Incremental Modelling and Solving.

Solving Computational Problems in the Theory of Word-Representable Graphs.

Cloud Benchmarking for Maximising Performance of Scientific Applications.

Conjure Documentation, Release 2.3.0.

Towards Improving Solution Dominance with Incomparability Conditions: A case-study using Generator Itemset Mining.

Modelling Langford's Problem: A Viewpoint for Search.

Memory Consistency Models using Constraints.

Automatically improving constraint models in Savile Row.

Extensible automated constraint modelling via refinement of abstract problem specifications.

Declarative Statistics.

Cloud Benchmarking For Maximising Performance of Scientific Applications.

The BIN\_COUNTS Constraint: Filtering and Applications.

Optimal Deployment of Geographically Distributed Workflow Engines on the Cloud.

Cloud Benchmarking for Performance.

Conjure Revisited: Towards Automated Constraint Modelling

>> java QueryDB 4

Publication venues in which "Özgür Akgün" has published in the last 3 years:

28th CP 2022: Haifa, Israel

18th CPAIOR 2021: Vienna, Austria

Artificial Intelligence, Volume 310

Mathematics of Computation, Volume 91

CoRR, February 2022

CoRR, May 2022

CoRR, November 2021

>> java QueryDB 5

Names of papers, and their venues that have been authored by “Özgür Akgün”:

Name of paper: A Framework for Generating Informative Benchmark Instances.

Venue: 28th CP 2022: Haifa, Israel

Name of paper: Understanding How People Approach Constraint Modelling and Solving.

Venue: 28th CP 2022: Haifa, Israel

Name of paper: Finding Subgraphs with Side Constraints.

Venue: 18th CPAIOR 2021: Vienna, Austria

Name of paper: Effective Encodings of Constraint Programming Models to SMT.

Venue: 26th CP 2020: Louvain-la-Neuve, Belgium

Name of paper: Discriminating Instance Generation from Abstract Specifications: A Case Study with CP and MIP.

Venue: 17th CPAIOR 2020: Vienna, Austria

Name of paper: Exploiting Incomparability in Solution Dominance: Improving General

Purpose Constraint-Based Mining. 24th ECAI 2020: Santiago de Compostela, Spain

Name of paper: Instance Generation via Generator Instances.

Venue: 25th CP 2019: Stamford, CT, USA

Name of paper: Automatic Streamlining for Constrained Optimisation.

Venue: 25th CP 2019: Stamford, CT, USA

Name of paper: Automatic Discovery and Exploitation of Promising Subproblems for Tabulation.

Venue: 24th CP 2018: Lille, France

Name of paper: Automatic Generation and Selection of Streamlined Constraint Models via Monte Carlo Search on a Model Lattice.

Venue: 24th CP 2018: Lille, France

Name of paper: Metamorphic Testing of Constraint Solvers.

Venue: 24th CP 2018: Lille, France

Name of paper: Closed Frequent Itemset Mining with Arbitrary Side Constraints.

Venue: 18th ICDM 2018: Singapore - Workshops

Name of paper: A Framework for Constraint Based Local Search using Essence.

Venue: 27th IJCAI 2018: Stockholm, Sweden

Name of paper: Using Metric Space Indexing for Complete and Efficient Record Linkage.

Venue: 22nd PAKDD 2018: Melbourne, VIC, Australia

Name of paper: Exploiting Short Supports for Improved Encoding of Arbitrary Constraints into SAT.

Venue: 22. CP 2016: Toulouse, France

Name of paper: Automatically Generating Streamlined Constraint Models with Essence and Conjure.

Venue: 21. CP 2015: Cork, Ireland

Name of paper: Cloud-based E-Infrastructure for Scheduling Astronomical Observations.

Venue: 11th eScience 2015: Munich, Germany

Name of paper: Cloud Benchmarking for Performance.

Venue: CloudCom 2014: Singapore

Name of paper: Optimal Deployment of Geographically Distributed Workflow Engines on the Cloud.

Venue: CloudCom 2014: Singapore

Name of paper: Automatically Improving Constraint Models in Savile Row through Associative-Commutative Common Subexpression Elimination.

Venue: 20. CP 2014: Lyon, France

Name of paper: Breaking Conditional Symmetry in Automated Constraint Modelling with CONJURE.

Venue: 21st ECAI 2014: Prague, Czech Republic

Name of paper: Automated Symmetry Breaking and Model Selection in Conjure.

Venue: 19. CP 2013: Uppsala, Sweden

Name of paper: Extensible Automated Constraint Modelling.

Venue: 25th AAAI 2011: San Francisco, California, USA

Name of paper: Conjure: Automatic Generation of Constraint Models from Problem Specifications.

Venue: Artificial Intelligence, Volume 310

Name of paper: Enumeration of set-theoretic solutions to the Yang-Baxter equation.

Venue: Mathematics of Computation, Volume 91

Name of paper: Automatic Tabulation in Constraint Models.

Venue: CoRR, February 2022

Name of paper: A Framework for Generating Informative Benchmark Instances.

Venue: CoRR, May 2022

Name of paper: Towards Reformulating Essence Specifications for Robustness.

Venue: CoRR, November 2021

Name of paper: How People Visually Represent Discrete Constraint Problems.

Venue: IEEE Transactions on Visualization and Computer Graphics, Volume 26

Name of paper: Towards Portfolios of Streamlined Constraint Models: A Case Study with the Balanced Academic Curriculum Problem.

Venue: CoRR, September 2020

Name of paper: Exploring Instance Generation for Automated Planning.

Venue: CoRR, September 2020

Name of paper: Efficient Incremental Modelling and Solving.

Venue: CoRR, September 2020

Name of paper: Solving Computational Problems in the Theory of Word-Representable Graphs.

Venue: Journal of Integer Sequences, Volume 22

Name of paper: Cloud Benchmarking for Maximising Performance of Scientific Applications.

Venue: IEEE Transactions on Cloud Computing, Volume 7

Name of paper: Conjure Documentation, Release 2.3.0.

Venue: CoRR, October 2019

Name of paper: Towards Improving Solution Dominance with Incomparability Conditions: A case-study using Generator Itemset Mining.

Venue: CoRR, October 2019

Name of paper: Modelling Langford's Problem: A Viewpoint for Search.

Venue: CoRR, August 2018

Name of paper: Memory Consistency Models using Constraints.

Venue: CoRR, August 2018

Name of paper: Automatically improving constraint models in Savile Row.

Venue: Artificial Intelligence, Volume 251

Name of paper: Extensible automated constraint modelling via refinement of abstract problem specifications.

Venue: Constraints - An International Journal, Volume 22

Name of paper: Declarative Statistics.

Venue: CoRR, August 2017

Name of paper: Cloud Benchmarking For Maximising Performance of Scientific Applications.

Venue: CoRR, August 2016



Name of paper: The BIN\_COUNTS Constraint: Filtering and Applications.

Venue: CoRR, November 2016

Name of paper: Optimal Deployment of Geographically Distributed Workflow Engines on the Cloud.

Venue: CoRR, October 2014

Name of paper: Cloud Benchmarking for Performance.

Venue: CoRR, November 2014

Name of paper: Conjure Revisited: Towards Automated Constraint Modelling

Venue: CoRR, September 2011

### **Evaluation**

I am proud to say that my program works: it can create the database, initialise it properly, populate it with data fetched from xml files (either from DBLP or from the cache), and query the database. The testing also works.

### **Conclusion**

All in all, I have found this assignment easier than the previous one. Firstly, I have already done some SQL before coming to St Andrews. Secondly, most of the code in the PopulateDB.java class was re-used from the previous practical. Even if JDBC was completely new to me (and I struggled a bit at the beginning), I was able to adapt and complete this CS10003 Practical 3.