28TH BRAZILIAN SYMPOSIUM ON DATABASES

TUTORIALS

PROCEEDINGS

September 30th – October 3rd, 2013
Recife, Pernambuco, Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Pernambuco – UFPE

Realization

Centro de Informática (CIn)

Tutorials Chair

Mirella M. Moro, UFMG

Editorial

The goal of the Tutorial Session of the Brazilian Symposium on Databases (SBBD) 2013 is to offer conference attendees a stimulating and informative selection of tutorials that disseminate knowledge related to the area of database systems. Each tutorial is presented by experts on its topics, and reflects the high academic and research standards of the SBBD conference. This special session includes introductory and advanced tutorials. Introductory tutorials are directed towards graduate students, last-year undergraduate students, as well as attendees from industry. They may be considered as recycling courses, although they do not concern traditional textbook topics. On the other hand, advanced tutorials cover the state-of-the-art of a given topic, motivating and exposing potential research issues.

This year, the SBBD Steering Committee selected three tutorials for presentation. The first tutorial, "*Solid-State Disks: How Do They Change the DBMS Game?*" (to be presented in Portuguese), is presented by Angelo Brayner and Mário A. Nascimento. It aims at bringing a new trend, SSD-based DBMS, and its implications to the attention of the Brazilian database community. The authors discuss different core DBMS techniques and algorithms through highlighting recent research on practical issues such as: indexing, join processing, query optimization, caching and logging. Angelo Brayner leads the CEARA (*advanCEd dAtabase Research*) research group at University of Fortaleza, Brazil. Mário A. Nascimento is a professor and associate chair (research) at the Department of Computing Science of the University of Alberta, Canada.

The second tutorial, "*Cloud Computing for Sciences: the Fundamental Role of Databases*" (to be presented in Portuguese), is presented by Daniel de Oliveira and Marta Mattoso. It covers the databases fundamentals that may be used for processing scientific data over workflows in the cloud. Daniel de Oliveira is professor of the Instituto de Computação da Universidade Federal Fluminense, Brazil. Marta Mattoso is professor at COPPE/Universidade Federal do Rio de Janeiro, Brazil.

The third tutorial, "*Querying Data through Ontologies*", is presented by Andrea Cali and Andreas Pieris. It overviews of the most important ontology formalisms for the Semantic Web and illustrates the most relevant query answering techniques, with particular emphasis on their efficiency. Andrea Cali is a Lecturer at the Department of Computer Science and Information Systems of the University of London, Birkbeck College, UK. Andreas Pieris is a post-doctoral researcher at the Department of Computer Science of the University of Oxford, UK.

On behalf of the SBBD Steering Committee, I would like to thank all tutorial authors for contributing with SBBD 2013.

I hope you all enjoy SBBD in Recife/PE.

Mirella M. Moro
SBBD 2013, Tutorials Chair

28TH BRAZILIAN SYMPOSIUM ON DATABASES

September 30th – October 3rd, 2013

Recife, Pernambuco, Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Pernambuco – UFPE

Realization

Centro de Informática (CIn)

SBBD Steering Committee

Marco A. Casanova, PUC-Rio (Chair)
Ana Carolina Salgado, CIn-UFPE
Cristina Dutra de Aguiar Ciferri, USP
José Palazzo Moreira de Oliveira, UFRGS
Mirella M. Moro, DCC-UFMG

SBBD 2013 Committee

Steering Committee Chair
Marco A. Casanova, PUC-Rio
Local Organization Chair
Bernadette Farias Lóscio, CIn-UFPE
Program Committee Chair
Cristina Dutra de Aguiar Ciferri, USP
Short Papers Chairs
Renato Fileto, UFSC and Marco Cristo, UFAM
Demos and Applications Chairs
Damires Souza, IFPB and Daniel Kaster, UEL
Thesis and Dissertation Workshop Chairs
Karin Becker, UFRGS and André Santanchè, UNICAMP
Tutorials Chair
Mirella M. Moro, DCC-UFMG
Lectures Chair
João Eduardo Ferreira, IME-USP

Local Organization Committee

Bernadette Farias Lóscio, CIn-UFPE (Chair)
Ana Carolina Salgado, CIn-UFPE
Agenor Marinho de Souza Neto, CIn-UFPE

Carlos Eduardo Santos Pires, DSC-UFCG
Danusa Ribeiro Cunha, CIn-UFPE
Priscilla Vieira, CIn-UFPE
Robson Fidalgo, CIn-UFPE

28TH BRAZILIAN SYMPOSIUM ON DATABASES

TUTORIALS

Table of Contents

Solid-State Disks: How Do They Change the DBMS Game?

Angelo Brayner[1], Mario A. Nascimento[2]

[1] University of Fortaleza, Brazil
[2] University of Alberta, Canada
brayner@unifor.br, mario.nascimento@ualberta.ca

It is well know that the speed of processors has increased exponentially whereas the number of inputs/outputs per second (IOPS) afforded by hard-disk drivers (HDDs) has only increased marginally. A set of new storage media, called Solid State Disk (SSD), has emerged as a promising solution to decrease the difference between an HDD's data access time and the time that processors can consume data [Boboila and Desnoyers 2011]. Regarding storage capacity, SSDs still have a capacity inferior to their HDD counterparts. However, in the current pace of SSD technology development, we can expect that in the near future SSD storage capacity might be similar to that provided by HDDs. For that reason, we can also expect that SSDs be used in large scale within database management systems (DBMSs). However, the *modus operandi* of SSDs are rather different when compared to HDDs, which motivates revisiting many well-established techniques and algorithms in the core of a DBMS; all originally oriented towards HDDs. This tutorial focuses exactly on such issues, aiming at bringing this new trend, SSD-based DBMS, and its implications to the attention of the Brazilian database community.

A typical SSD is a computer chip which can be electrically reprogrammed and erased. An SSD stores data in an array of floating-gate transistors, called cells. Bits are represented by means of the voltage level in a cell. A cell with high voltage level represents a bit 1 (default state), whereas a low voltage level represents a bit 0. Sets of cells form data pages, data pages are organized in blocks and, finally, sets of blocks are stores within chips. There are three operations which can be executed on a flash device: read, erase and program [Kim and Koh 2004]. A *read* operation may randomly occur anywhere in a flash device. An *erase* operation is applied to a whole block, i.e., a set of pages, setting all bits to 1. A *program* operation sets a bit to 0. It is important to note that a program operation can only be executed on a "clean" (free) block, which is a block with all bits set to 1. Since SSDs have their lifetime determined by the number of write operations, a technique called *wear leveling* is applied to prolong its useful life. The key goal is to evenly spread out write operations across the storage area of the medium.

One of the most evident feature of SSD technology is the absence of mechanical parts in their assembly, only semi-conductors (chips) are used. Due to such a feature, SSDs presents a few interesting characteristics, for example:

(1) Low energy footprint. This is because, to perform read/write operations, there are only logical gates (circuitry) are involved;

(2) Low random access time. SSDs allows random access at least a few orders of magnitude faster than hard disk drives (HDDs);

(3) High random IOPS rates. Since SSDs have no mechanical moving parts, there is no mechanical seek time or latency to overcome.

(4) Asymmetry of read and write execution time. As a consequence of the technology used in SSDs, a read operation may be 1-2 orders of magnitude faster than a write operation.

From a database perspective, the first three characteristics are directly beneficial to existing database systems. On the other hand, the last characteristic, read/write asymmetry, poses new challenges to database technology, since write-intensive components (e.g., query processing and recovery components) of a database system (DBMS) may hinder a SSD-based DBMS performance, given they were designed for a symmetric rather than an asymmetric I/O (sub-)system.

In this tutorial we will discuss a number of core DBMS techniques and algorithms and will highlight recent research, e.g., [Tsirogiannis et al. 2009; Graefe et al. 2010; Fang et al. 2011; Sarwat et al. 2011; Koltsidas and Viglas 2011] that has addressed them in the context of this HDs/SSDs change of paradigm, thus highlighting opportunities for relevant and practical research, for instance:

—Indexing. It is known that conventional B+-trees may have high write-operation rates due to so-called "node splits". Some techniques to overcome such a problem will be discussed.
—Join Processing. Classical join physical operators require temporary result storage, i.e., they may increase write-operation rates during query processing. New join physical operators which reduce the number write-operations will be analyzed.
—Query optimization. Conventional query optimization techniques aim at reducing the number of disk pages transferred between disk and buffer area. While reading data stored in SSDs is not a problem anymore, writing pages into SSDs may negatively impact the DBMS's IOPS rate. In this sense, new ideas, in particular new query-execution cost models will be discussed.
—Caching. An SSD-aware database buffer management policy may reduce the impact of write operations on SSD-based DBMSs, hence SSD-aware buffer replacement policies will be discussed as well.
—Logging. Methods to delay log recording (e.g., a "batch" logging) could help to alleviate the write overhead inherent to SSDs. Other research thread to be discussed in this topic is the use of different log file structures, which may take advantage of the SSD behaviour. New commit processing techniques adapted to SSDs will also be discussed.

Acknowledgements: M.A. Nascimento has been partially supported by NSERC, Canada, and is also a visiting Professor at Federal University of Ceará, Brazil and Ludwig-Maximilians-Universität München, Germany

REFERENCES

Boboila, S. and Desnoyers, P. Performance models of flash-based solid-state drives for real workloads. In *Proc. of the IEEE 27th Symposium on Mass Storage Systems and Technologies*, A. Brinkmann and D. Pease (Eds.). IEEE Computer Society, pp. 1–6, 2011.

Fang, R., Hsiao, H.-I., He, B., Mohan, C., and Wang, Y. High performance database logging using storage class memory. In *Proc. of the 27th IEEE Intl. Conf. on Data Engineering*, S. Abiteboul, K. Böhm, C. Koch, and K.-L. Tan (Eds.). IEEE Computer Society, pp. 1221–1231, 2011.

Graefe, G., Harizopoulos, S., Kuno, H. A., Shah, M. A., Tsirogiannis, D., and Wiener, J. L. Designing database operators for flash-enabled memory hierarchies. *IEEE Data Eng. Bull.* 33 (4): 21–27, 2010.

Kim, K. and Koh, G.-H. Future memory technology including emerging new memories. In *Proc. of the 24th International Conference on Microelectronics*. Vol. 1. IEEE, pp. 377–384, 2004.

Koltsidas, I. and Viglas, S. Data management over flash memory. In *Proc. of the 2011 ACM SIGMOD Intl. Conf. on Management of Data*, T. K. Sellis, R. J. Miller, A. Kementsietsidis, and Y. Velegrakis (Eds.). ACM, pp. 1209–1212, 2011.

Sarwat, M., Mokbel, M. F., Zhou, X., and Nath, S. Fast: A generic framework for flash-aware spatial trees. In *Proc. of the 12th Symp. on Advances on Spatial and Temporal Databases*, D. Pfoser, Y. Tao, K. Mouratidis, M. A. Nascimento, M. F. Mokbel, S. Shekhar, and Y. Huang (Eds.). Lecture Notes in Computer Science, vol. 6849. Springer, pp. 149–167, 2011.

Tsirogiannis, D., Harizopoulos, S., Shah, M. A., Wiener, J. L., and Graefe, G. Query processing techniques for solid state drives. In *Proc. of the 2009 ACM SIGMOD Intl. Conf. on Management of Data*, U. Çetintemel, S. B. Zdonik, D. Kossmann, and N. Tatbul (Eds.). ACM, pp. 59–72, 2009.

Computação em Nuvem para Ciência: o Papel Fundamental da Área de Bancos de Dados

Daniel de Oliveira[1], Marta Mattoso[2]

[1] Instituto de Computação - Universidade Federal Fluminense
[2] COPPE - Universidade Federal do Rio de Janeiro
danielcmo@ic.uff.br, marta@cos.ufrj.br

O uso de computação em nuvem [Vaquero et al. 2009] para apoiar o desenvolvimento da ciência é um tema complexo que conta com linhas de financiamento específicas em agências de fomento de vários países. Esse apoio envolve diversas disciplinas da computação como o processamento paralelo, redes de sensores, visualização e principalmente bancos de dados. A característica principal do apoio ao desenvolvimento da ciência em larga escala passa pela gerência de fluxos de dados científicos (*i.e.* *data-flow* em *workflow*s) [Taylor et al. 2007][Mattoso et al. 2010] em grande volume. Assim como os sistemas de bancos de dados surgiram para gerenciar dados, acima dos serviços básicos de gerência de arquivos do sistema operacional, os sistemas de gerência de *workflow*s orientados ao fluxo de dados científicos (SGWfC) desempenham esse mesmo papel hoje. Ou seja, os SGWfC tiram partido do conhecimento do fluxo de dados para realizar um escalonamento de recursos orientado ao *workflow* científico, para gerenciar réplicas de dados, manter os dados consistentes por meio de recuperação de falhas, gerir o acesso concorrente, dentre outros.

A área de Bancos de Dados vem usando seus fundamentos para gerenciar grandes conjuntos de dados ao longo de décadas, usando, por exemplo, processamento paralelo e distribuído. Um exemplo é a otimização do *workflow*. Assim como a álgebra relacional possibilita a otimização do plano de execução de uma consulta, uma álgebra voltada ao fluxo de dados do *workflow* científico [Ogasawara et al. 2011] também pode ser usada para otimizar o plano de execução de um *workflow* orientado a dados. Usar um SGBD tradicional para dados científicos não é a solução, uma vez que SGBDs são voltados para dados da área de negócios. Por outro lado, ignorar as contribuições de fundamentos de bancos de dados na gerência de fluxos de dados e criar soluções visando apenas à eficiência da programação funcional (*e.g.* o modelo Map-Reduce) ou ao escalonamento de recursos no nível de infraestrutura de nuvem, pode levar a soluções limitadas ou a redescobrir algoritmos e tecnologias. A execução de *workflow*s em nuvens de computadores apresenta vários desafios. Por exemplo, como os próprios cientistas executam os *workflow*s, é difícil decidir *a priori* a quantidade de recursos e por quanto tempo os mesmos serão necessários ou como distribuir as tarefas nas diversas máquinas virtuais durante a execução do *workflow*. Usar serviços de nuvens que desconhecem a natureza do fluxo de dados de *workflow*s científicos pode levar ao uso inadequado dos recursos.

Além da gerência da execução paralela, os cientistas têm de gerenciar outras questões, como a captura de proveniência [Freire et al. 2008] distribuída de forma a garantir a validade, reprodutibilidade e a análise do *workflow*. Os resultados de um *workflow* científico devem ser passíveis de reprodução para serem considerados válidos cientificamente. Tal reprodução é fortemente baseada no uso de dados de proveniência. A proveniência lida com os dados históricos do *workflow*, sejam eles relativos à sua estrutura (*i.e.* prospectiva) ou a sua execução (*i.e.* retrospectiva). Além de apoiar a reprodução, os dados de proveniência podem oferecer subsídios para resolver muitos dos problemas em aberto na execução de *workflow*s em nuvens de computadores como o escalonamento de atividades, tolerância

a falhas e monitoramento [Oliveira et al. 2012]. Como esses dados de proveniência podem ser representados de diferentes formas (modelo relacional, XML, grafo, etc.) que são usualmente apoiadas por SGBD, podemos nos beneficiar de técnicas de banco de dados já existentes. Além disso, a utilização da proveniência no apoio à gerência deste tipo de experimento nos traz novas oportunidades na área de banco de dados, seja na representação e captura dos dados, na otimização da consulta submetida pelo cientista, na fragmentação dos dados científicos e de proveniência, etc.

Nesse tutorial, abordamos como fundamentos de bancos de dados podem ser usados no processamento de dados científicos em sintonia com os SGWfC em nuvens de computadores. Consideramos que essa combinação está no caminho crítico do apoio ao desenvolvimento de ciência em larga escala em nuvens computacionais. Mostraremos um panorama do estado da arte no uso da computação em nuvem para apoiar o desenvolvimento da ciência computacional. Apresentaremos as soluções principais nesse apoio em nuvens, a saber, Pegasus [Deelman et al. 2004], Swift [Wilde et al. 2009], Tavaxy [Abouelhoda et al. 2012], Nephele [Warneke and Kao 2009] e SciCumulus [Oliveira et al. 2010], destacando suas principais contribuições como infraestrutura. Discutiremos as oportunidades de pesquisas em bancos de dados quanto à gerência de dados e processos científicos, aos aspectos de distribuição de dados e atividades dos *workflow*s, ao acompanhamento da execução de longa duração por parte do cientista, à gerência de dados de proveniência e à combinação de dados de proveniência com dados científicos do domínio da aplicação.

REFERENCES

Abouelhoda, M., Issa, S., and Ghanem, M. Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. *BMC Bioinformatics* 13 (77): 1–19, 2012.

Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M.-H., Vahi, K., and Livny, M. Pegasus : Mapping Scientific Workflows onto the Grid. In *Proceedings of the Across Grids Conference*, 2004.

Freire, J., Koop, D., Santos, E., and Silva, C. Provenance for Computational Tasks: A Survey. *Computing in Science and Engineering* 10 (3): 11–21, 2008.

Mattoso, M., Werner, C., Travassos, G. H., Braganholo, V., Murta, L., Ogasawara, E., de Oliveira, D., Cruz, S. M. S., and Martinho, W. Towards Supporting the Life Cycle of Large-scale Scientific Experiments. *International Journal of of Business Process Integration and Management* 5 (1): 79–92, 2010.

Ogasawara, E., Dias, J., de Oliveira, D., Porto, F., Valduriez, P., and Mattoso, M. An Algebraic Approach for Data-Centric Scientific Workflows. *Proceedings of the International Conference on Very Large Data Bases* 4 (1): 1328–1339, 2011.

Oliveira, D., Ocana, K., Baiao, F., and Mattoso, M. A Provenance-based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds. *Journal of Grid Computing* 10 (1): 521–552, 2012.

Oliveira, D., Ogasawara, E., Baiao, F., and Mattoso, M. SciCumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows. In *Proceedings of the 2010 IEEE Cloud Computing Conference*, 2010.

Taylor, I., Deelman, E., Gannon, D., and Shields, M. *Workflows for e-Science: Scientific Workflows for Grids*. Springer Verlag, 2007.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun.* 39 (1): 50–55, 2009.

Warneke, D. and Kao, O. Nephele: Efficient Parallel Data Processing in the Cloud. In *Proceedings of the Many Task Computing Workshop*, 2009.

Wilde, M., Hategan, M., Wozniak, J., Clifford, B., Katz, D., and Foster, I. Swift: A language for distributed parallel scripting. *Parallel Computing* 1 (1): 633–652, 2009.

Querying Data through Ontologies

Andrea Calì[1], Andreas Pieris[2]

[1] Birkbeck, University of London
[2] University of Oxford
andrea@dcs.bbk.ac.uk, andreas.pieris@cs.ox.ac.uk

An *ontology* is an explicit specification of a conceptualization of an area of interest, and consists of a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. The need for semantically enhancing existing databases with ontological onstraints gave rise to the so-called *ontological database management systems*, that is, a new type of DMBSs equipped with advanced reasoning and query processing mechanisms. In particular, an extensional database D is combined with an ontology Σ which derives new intensional knowledge from D. An input query is not just answered against the database, as in the classical setting, but against the logical theory (a.k.a. ontological database) $D \cup \Sigma$. Database technology providers have recognized the need for combining ontological reasoning and database technology, and have recently started to build ontological reasoning modules on top of their existing software with the aim of delivering effective database management solutions to their customers. For example, Oracle Inc. offers a system, called Oracle Database 11g, enhanced by modules performing ontological reasoning tasks[1]. Enhancing databases with ontologies is also at the heart of several research-based systems such as QuOnto [Acciarri et al. 2005].

Description Logics (DLs) [Baader et al. 2003] are a family of knowledge representation languages widely used in ontological modeling. In fact, DLs model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations on classes of individuals, respectively. Interestingly, DLs provide the logical underpinning for the Web Ontology Language (OWL), and its revision OWL 2, as standartized by the W3C[2]. However, in order to achieve favorable computational properties, DLs are able only to describe knowledge for which the underlying relational structure is treelike. Moreover, they usually support only unary and binary relations. Recently, the Datalog$^{\pm}$ family [Calì et al. 2012] of ontology languages has been proposed, with the purpose of overcoming the above limitations of DLs. Datalog$^{\pm}$ languages are based on Datalog rules that allow for the existential quantification of variables in the head, in the same fashion as Datalog with *value invention* [Cabibbo 1998]. The absence of value invention, thoroughly discussed in [Patel-Schneider and Horrocks 2007], is the main shortcoming of Datalog in modeling ontologies. The basic Datalog$^{\pm}$ rules are (function-free) Horn rules extended with existential quantification in the head, known as *tuple-generating dependencies* or *existential rules*. The addition of *negative constraints* of the form $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \bot$, where $\bot$ denotes the truth constant *false*, and of restricted classes of *equality-generating dependencies* such as *key dependencies*, makes Datalog$^{\pm}$ expressive enough to capture the most common tractable ontology languages such as the *DL-Lite* family of DLs [Calvanese et al. 2007].

Query answering under Datalog rules extended with existential quantification in the head is undecidable (implicit in [Beeri and Vardi 1981]). Therefore, some syntactic restriction is needed to ensure decidability (hence the symbol "$\pm$"). The fundamental restriction paradigms that have been studied so far are as follows:

[1] http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html
[2] http://www.w3.org/TR/owl2-overview/

Weak-Acyclicity. Weakly-acyclic Datalog$^{\pm}$, introduced in the context of data exchange [Fagin et al. 2005], guarantees the finiteness of the universal model of the given ontological database. Notice that a universal model (a.k.a. canonical model) acts as a representative of all the models of the given ontological database, and thus for query answering purposes we can consider only this special model. Therefore, one can just evaluate the given query over the finite universal model.

Guardedness. Decidability of query answering under guarded Datalog$^{\pm}$ follows from the fact that guardedness ensures the existence of a universal model of finite treewidth. Guarded Datalog$^{\pm}$ rules and generalizations thereof were studied in [Calì et al. 2013; Calì et al. 2012; Baget et al. 2011]. Notice that sets of guarded Datalog$^{\pm}$ rules can be rewritten as theories in the guarded fragment of first-order logic [Andréka et al. 1998].

Stickiness. Sticky Datalog$^{\pm}$ has been proposed in [Calì et al. 2012] with the aim of identifying an expressive language that allows for joins in rule-bodies which are expressible only via non-guarded rules. Stickiness enures the termination of resolution-based procedures, which in turn implies that we can construct the (finite) part of the universal model which is needed in order to entail the query.

Shyness. Shy Datalog$^{\pm}$ ensures that during the construction of the universal model only database onstants can participate in a join operation [Leone et al. 2012]. This property allows us to consider only a finite part of the (possibly infinite) universal model which can be effectively constructed.

Towards the identification of even more expressive languages, several attempts have been conducted to consolidate the aforementioned paradigms. Notable formalisms are *glut-guardedness* [Krötzsch and Rudolph 2011] and *weak-stickiness* [Calì et al. 2012], obtained by joining weak-acyclicity with guardedness and stickiness, respectively. A condition, called *tameness*, which allows for a safe combination of guardedness and stickiness has been recently proposed in [Gottlob et al. 2013].

Acknowledgments. Andrea Calì acknowledges support by the EPSRC project "Logic-based Integration and Querying of Unindexed Data" (EP/E010865/1).

REFERENCES

Acciarri, A., Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., Palmieri, M., and Rosati, R. QuOnto: Querying ontologies. In *Proc. of AAAI*. pp. 1670–1671, 2005.

Andréka, H., van Benthem, J., and Németi, I. Modal languages and bounded fragments of predicate logic. *J. Phil. Log.* vol. 27, pp. 217–274, 1998.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

Baget, J.-F., Leclère, M., Mugnier, M.-L., and Salvat, E. On rules with existential variables: Walking the decidability line. *Artif. Intell.* 175 (9-10): 1620–1654, 2011.

Beeri, C. and Vardi, M. Y. The implication problem for data dependencies. In *Proc. of ICALP*. pp. 73–85, 1981.

Cabibbo, L. The expressive power of stratified logic programs with value invention. *Inf. Comput.* 147 (1): 22–56, 1998.

Calì, A., Gottlob, G., and Kifer, M. Taming the infinite chase: Query answering under expressive relational onstraints. *J. Artif. Intell. Res.*, 2013. To appear.

Calì, A., Gottlob, G., and Lukasiewicz, T. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* vol. 14, pp. 57–83, 2012.

Calì, A., Gottlob, G., and Pieris, A. Ontological query answering under expressive entity-relationship schemata. *Inf. Syst.* 37 (4): 320–335, 2012.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. Tractable reasoning and efficient query answering in description logics: The DL-lite family. *J. Autom. Reasoning* 39 (3): 385–429, 2007.

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. Data exchange: Semantics and query answering. *Theor. Comput. Sci.* 336 (1): 89–124, 2005.

Gottlob, G., Manna, M., and Pieris, A. Combining decidability paradigms for existential rules. *TPLP*, 2013. To appear.

Krötzsch, M. and Rudolph, S. Extending decidable existential rules by joining acyclicity and guardedness. In *Proc. of IJCAI*. pp. 963–968, 2011.

Leone, N., Manna, M., Terracina, G., and Veltri, P. Efficiently computable Datalog$^{\exists}$ programs. In *Proc. of KR*, 2012.

Patel-Schneider, P. F. and Horrocks, I. A comparison of two modelling paradigms in the semantic web. *J. Web Sem.* 5 (4): 240–250, 2007.