Today's industrial-strength database engines, both commercial and public-domain, are designed to provide highly sophisticated functionalities, making them the backbone of the information society. Not surprisingly, a fallout of this sophistication is that the internal software infrastructure has become extremely complex, making it a technically challenging task to (a) verify the correctness of the engine components, and (b) tune the system to meet the desired performance objectives.

This issue of the Data Engineering Bulletin describes both novel research proposals and current industrial practices for the testing and tuning of enterprise database systems. Historically, these topics have received comparatively little attention in the research literature. However, there has been growing awareness in recent years of the rich set of problems on offer, which are simultaneously technically challenging and of immediate practical relevance, as exemplified by the articles featured in this issue.

The first article, by Debnath, Mokbel and Lilja from U. of Minnesota, considers the problem of efficently selecting, from among a dauntingly large number, the most relevant configuration parameters for tuning a database engine. They employ an experimental design methodology that makes the computational effort linear in the number of parameters, and quantitatively demonstrate with an implementation on PostgreSQL, that this approach is capable of successfully identifying the critical parameters.

The second article by Mehta, Gupta, Wang and Dayal of HP Labs presents a holistic machine-learning-based approach towards workload management for enterprise data warehouses addressing issues of admission control, scheduling and progress monitoring. Their techniques have been evaluated on real-world warehousing environments with promising results.

The third article by Krompass, Scholz, Albutiu and Kemper from TU Munich, and Kuno, Wiener and Dayal from HP Labs, investigates the specification and satisfaction of quality-of-service objectives in the context of operational data stores hosting workloads with a mix of transactional and decision support queries. They also survey the infrastructure provided by current industrial products to support these objectives.

The fourth article by Binnig, Kossmann and Lo from SAP, ETH and Hong Kong Polytechnic U., respectively, brings a fresh outlook to the design of test databases through the use of database techniques such as declarative specifications and logical data independence. Specifically, they advocate the ideas of "reverse query processing" (given a schema, a query and a result, generate a compliant database), and "symbolic query processing" (the database consists of symbolic, rather than concrete, data), which can be used to test engine components.

We then have a series of articles on current industry practices. First, Giakoumakis and Galindo-Legaria from Microsoft provide a guided tour through the arduous world of testing database query optimizers. They also overview the array of techniques used in testing the SQL Server optimizer. Then, Joshi, Lamb and Sandstrom from Oracle present the testing tools utilized for the Berkeley DB family of database engines, their task rendered additionally difficult because users, taking advantage of the source-code availability, may either modify or port the engines to new platforms. Next, Yagoub, Belknap, Dageville, Dias, Joshi and Yu from Oracle present the SQL performance analyzer implemented in Oracle 11g to help users investigate "what-if" scenarios by forecasting and analyzing the impact of system changes on SQL workload performance before deployment. Finally, Gittens, Gupta, Godwin, Pereyra and Riihimaki of IBM, tackle the notoriously tricky problem of catching timing-related errors and defects in complex multi-threaded systems such as database engines. Their proposed technique attempts to trigger unexpected behavior by iteratively executing system tests with a background workload.

In closing, we thank all the article authors for their painstaking and timely efforts in developing their contributions for this special issue. Our hope is that the work presented here will serve as a strong stimulus for the academic and industrial research communities to address, with renewed vigor and resources, the development of stable and efficient database engines.