2007

Evolution of a Database Security course: using non-enterprise teaching tools

Justin Brown
Edith Cowan University

DOI: 10.4225/75/57b52b0443e2f
Originally published in the Proceedings of 5th Australian Information Security Management Conference, Edith Cowan University, Perth Western Australia, December 4th 2007.

Evolution of a Database Security course: using non-enterprise teaching tools

Justin Brown
Edith Cowan University
Perth, Western Australia
j.brown@ecu.edu.au

Abstract

This paper examines the issues in delivering a university unit of teaching in database security, examining problems in database environment selection and the ability to provide hands on training for students via on-campus and online modes. Initial problems with Linux and then Windows based enterprise database environments prompted the adoption of Microsoft Access as a database tool that was easier to deliver in-class and online. Though Access is file based and has fundamental flaws in its security implementation (within the enterprise context) it can be tweaked to emulate RDBMS level security, allowing students to see how a properly designed security model should operate. The paper shows that Microsoft Access can emulate field-level security with a correctly designed table and user model, but that the database itself should only be used to 'show and tell' security implementations, not apply them.

Keywords

Database security – elearning – RDBMS – Information security

INTRODUCTION

Database security is considered a required topic in any computer or information science course (Guimaraes, 2006; Srinivasan & Anup, 2005). Typically database security courses run at college or university level examine the various security models that can be applied to large SQL based RDBMS environments. Within such a course the students would be exposed to scenarios where various security models may be used, what weaknesses may be inherent in the model, and what the consequences of poor database security might be from the technical, legal and financial perspective. While the literature indicates that database security courses tend to be heavy on theory, they must also offer students some practical, hands-on database experience in order to reinforce of the paper based conceptual work.

This paper will examine the ongoing re-development of a university based third year unit (or course) of teaching which focuses on database security, referred to here as DB Sec. The term 'ongoing re-development' is used as the unit was initially given to this author to teach in 2003, at which time the conceptual structure of the unit was sound, but the practical teaching environment was dated. Oracle was in the process of being phased out of the teaching environment due to various hardware, software and licensing issues. At the same time the unit was required to be delivered in a mixed mode, being available to students both on-campus and online via the web.

THEORY, PRACTICE AND ASSESSMENT

At the time that this author took over the DB Sec unit the conceptual structure of the unit was quite strong, following a relatively 'classical' structure for a unit of study looking at database security (Binto & Anna, 2006). As with other university CS structures DB Sec was considered an 'advanced' (Udoh, 2006) database unit, in that it followed on from a more introductory second year unit in which students actually learned relational design and query language. Figure 1 shows the original breakdown of the 12 week teaching structure for DB Sec. Each week (or Module) contained a lecture, required readings and a set of Oracle 8i practical activities. The lecture materials and readings addressed the 'theory' aspect of the unit, with the required text being the seminal Database Security by Maria Fugini. The practice was largely composed of Oracle training materials and developers guides to Oracle 8i, with some interactive materials driven by some Oracle packages in Thomson's Net G online training suite. Assessments consisted of a 4000 word analytical report on a given database security scenario, with a requirement for a class presentation along with the report. The second assignment was the completion of the weekly practical lab materials to the satisfaction of the unit lecturer/tutor.

Module #	Title
1	Introduction, Aims of Security
2	Database Security Models
3	Database Security Policies and Plans
4	Database Risks and Security Issues
5	Database Management Systems and Oracle
6	Database Backup and Recovery
7	Database Access Controls, User Accounts
8	Client/Server & Distributed Database...
9	Database Auditing & Intrusion Detect...
10	Database Security and Internet
11	Data Mining and Data Warehousing
12	Security of Statistical Databases
13	Unit Review

Figure 1: Original unit structure for DB Sec

INITIAL CHANGES AND TEETHING PROBLEMS

The theory aspect of the unit was kept largely the same, though any lecture and reading materials that referred to specific database solutions were removed from the teaching program in favour of a more 'vanilla' context. The assessment component of the unit was changed to a more practical focus, in that the original written and presented assessment remained, though the second assessment specifically required students to deliver an implemented 'slice' of the system the proposed in their first assessment. For example, the first assessment in 2003 looked at the transfer and enhancement of a fictional university database system from one database environment to another. Students were required to look at the risks, data porting issues, security model and testing regime in such a scenario, as well as provide a template relational design for the core security structures. In the second assignment students had to design and implement a working portion, or slice of the overall system developed in the written assessment.

The one major change that could not be avoided was that of the practical aspect of the unit, with Oracle no longer available being available to the lab component of the semester structure. At the time the author wished to use MS Sql Server 2000, but given the security structure of the student lab environment, while MS SQL Server 2000 was available, students did not have admin privileges on the system, so could not create users and policies and apply them to specific databases. MySQL was briefly considered, but was also known to have security issues in the heavily managed lab environment, along with a lack of support for Views. In the end Postgres running on linux machines in the former Oracle lab was adopted, with extremely poor results, due to a number of reasons;

1. students had root level access to the linux machines as they were required to configure and run various linux distributions as a part of another 3rd year unit. As a result, by Thursday of the teaching week, the machines were usually unbootable and would require re-imaging, a process that could take 30 minutes.
2. Students were also struggling to replicate the lab environment at home, something which this author feels is critical to students being able to fully engage with a technology outside of the class, especially for online students (Brown, 2006).
3. A cd-rom bootable distribution of the linux environment, inclusive of a Postgres install with a third party GUI tool were given to students, though again, this proved too unreliable for students to use to a doable level.

Essentially no workable assignments were delivered using this environment and a decision was made to move away from the enterprise level RDBMS solutions in favour of a more flexible teaching tool.

MICROSOFT ACCESS AND THE PROOF OF CONCEPT APPROACH

After the initial problems with database environments in the delivery of DB Sec in 2003, it was decided that in 2004 a more flexible database environment needed to be used in the practical aspect of the unit, even if that database fell well short of a typical enterprise solution. The literature indicates that typical enterprise level databases used for the practical teaching aspects of database security courses range from Oracle or MS Sql (Binto & Anna, 2006; Bullers, Burd, & Seazzu, 2006) Server more recently back to Ingress in the late 1980's (Haas, 1988). While moving to Microsoft Access 2003 as the database teaching tool had issues with maintaining 'real-world' context, the decision was based on a number of factors;

1. students had access to the software in 90% of the computer labs within the school
2. students could download and install Access under the school's site license, allowing the online students to use the software if they did not already have it on their home machines
3. no specialized setup was required, nor was there a need for specific operating systems that normally run large RDBMS applications.

Further benefits of Access included its use of Structured Query Language, and its support for pre-defined Queries, giving students a near equivalent of Views as used in most large database solutions. In particular, Views are extremely important conceptually and practically in the field of database security as they allow for abstraction of original data structures with the benefit of read-only access. Obviously the main issue with Microsoft Access is that it is file rather than server based system, thus is inherently insecure in any meaningful way. Students were informed of this from the outset, though as 3rd year students most were aware of Access as a true secured database system. However, students were told that they could use Access to mimic a secured system to a significant degree, and that it was an excellent tool for displaying a 'proof of concept' regarding a given database security model

The literature on database security topics taught at tertiary institutions seems to indicate that the practical, hands on concepts students are taught are either account creation and data management (Bertino & Sandhu, 2005), or intrusion detection and threat reduction (Guimaraes, Mattord, & Austin, 2004). DB Sec adopted the former of these two approaches, specifically looking at role-based security applications within a database environment, thus exposing students to the middle ground between discretionary and mandatory access control regimes. As before, students were required to deliver a working proof of concept of the results of their first written assignment by providing a database solution in the second assignment. As a part of the teaching program, students were provided with an example database that was not related to their assignment topic, this database containing just the basic tables of the fictitious university used in the 2003 running of the unit (see Figure 2).

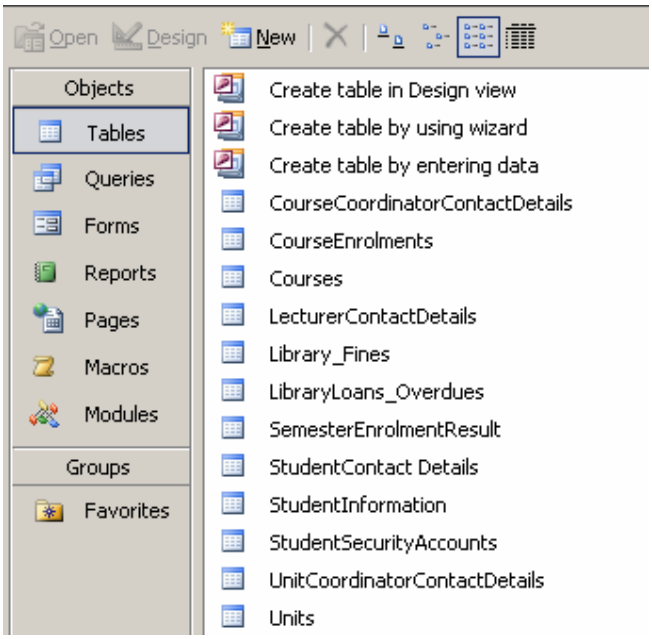

Figure 2: Basic table structure of example database

This database was designed to be relatively basic in structure, and not to necessarily represent a true enterprise implementation, which could contain dozens or hundreds of database objects (Tables, Views and Stored Procedures). The purpose of this database was as an exemplar to take students through the three main steps of producing a proof of concept security system in Microsoft Access;

1. create views (or queries in Access parlance) that represent objects that different user roles would be allowed access to
2. setup Access's Workgroup Administrator system and create individual users and user roles within it
3. assign the roles to the various objects in the database, such as tables, queries and forms so that only users within those roles have access to those objects

To achieve these three steps the lab teaching program was delivered in an interleaved approach, where students would use the example database to practice a concept one week, after which they would apply that skill on their assignment specific database the following week. This allowed in-class students to gain assistance with their assignment work from their lecturer or tutor, though only if they could demonstrate that they had made every effort to complete the previous week's materials. Online students were asked to send in their completed example database to let staff know that they were successfully engaging with the teaching materials. Online students who had shown effort on the example labs could then send in their assignment databases for assistance if they ran into troubles. Using this approach, students should have received three practical labs on different aspects of building role-based security in Microsoft Access, plus three assignment based labs where they were able to apply those skills directly to their assessable work.

EMULATING SECURITY IN ACCESS

Microsoft Access has a number of approaches for implementing security, ranging from setting a database password which allows any user in with admin rights as long as they know the password, through to user and group level rights applicable to different objects within a database. For DB Sec students were taken through the process of creating user and group level security structures, with users assigned to groups, and the groups representing logical roles. The roles specified for the students in the exemplar database included course coordinators, unit coordinators, lecturers, admin staff, library staff, IT support staff and of course, students.

The three main requirements asked of students in their security implementation were;

1) logged-in users could only access objects they had privileges on
2) logged-in users could only see/read their own data within a given table
3) logged-in users could edit certain fields, but not everything, such as a student or staff member being able to change their password, or contact details, but not their email address or staff/student id

These requirements could be considered relatively typical of any secured environment, particularly within a managed application. To achieve this level of security fidelity in Access a number of initial steps are required. Firstly a large number of queries, representing enterprise system views need to be established, each query focusing on a particular role-specific function, for example, student changing their password. In this example, a query would combine the student_details table (inclusive of student id, first name and surname) with the student_security table, which contains the student id and the student password, creating a tuple with the necessary elements for a password to be changed. Student level users would have Read Only (RO) level access to the student_details table and Read/Update (RU) access to the student_security table, which when combined in the change_student_password query would allow only the password to be changed, this change taking place directly into the student_security table (see Figure 3).

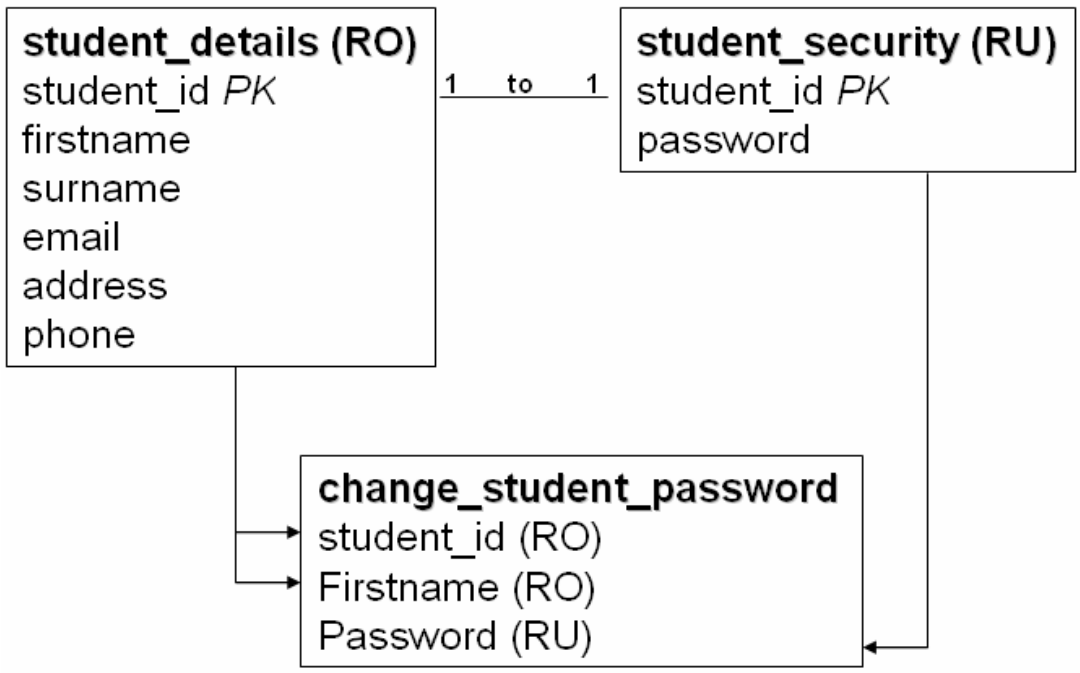

Figure 3: Example of a field level security in an Access data structure

Obviously the logical problem with such an example is that in most environments Views are designed as truly read only, and are not used for conducting updates such as in this example. However, Microsoft Access will allow this to happen, as long as the Recordset Type in the Query is set to Dynaset (Inconsistent Updates) and the query itself is given Read/Update rights and only contains two source tables. Staff would have a similar structure that allows them to change their own password, with this concept being extended to cover any set of Read, Update, Insert or Delete (in Access parlance) functions for a given user and task. Essentially, in order to reflect how a secured environment should operate, Microsoft Access needs tables, table data, queries and user/group settings to all be aligned.

Students are taught to create their database, and then create queries based on functions that the end users may wish to perform (such as the change password example) or perhaps just data that users may need to read (such as student specific enrolments). After this students follow the User Level Security Wizard in Microsoft Access that takes them through setting up the external Workgroup security file (a .MDW file that becomes associated with the main Access .MDB file) and creating one overall admin level account. During this process the database security system is set to allow individual users no permissions on any database objects, short of actually opening the database itself (see Figure 4).

This is basically a deny-all access model where individual users have no rights beyond loading the database file itself. If a user is not assigned to a group, they will not be able to perform any actions within the database or access any objects. Once this wizard is completed, students log into the database using the admin level account they created, then go through the process of adding user accounts for both students and staff, those accounts having usernames that match the student and staff id's in the tables of the database. Once this process is completed, students then add the various groups described earlier, after which they assign the various users to their allotted groups (see Figure 5).

Finally, students then go through the process of assigning the groups to database objects, specifically the tables and queries, specifying what level of permissions each group will have on each object (see Figure 6).

To round out the security emulation in Microsoft Access students are shown how to link logged-in users to their own data using the *CurrentUser()* function in Access (see Figure 7).

In this case students learn that if they have a user account with the same id as a staff or student id used to generate any of the many user-specific queries, then that user id can be automatically fed into the query so only that user's data is displayed. Students are also taken through the process of setting up forms, then menu switchboards which automatically load when the database does. Students are encouraged to 'hide' the original database tables and queries, only providing access to objects via the visual switchboard/form interface.

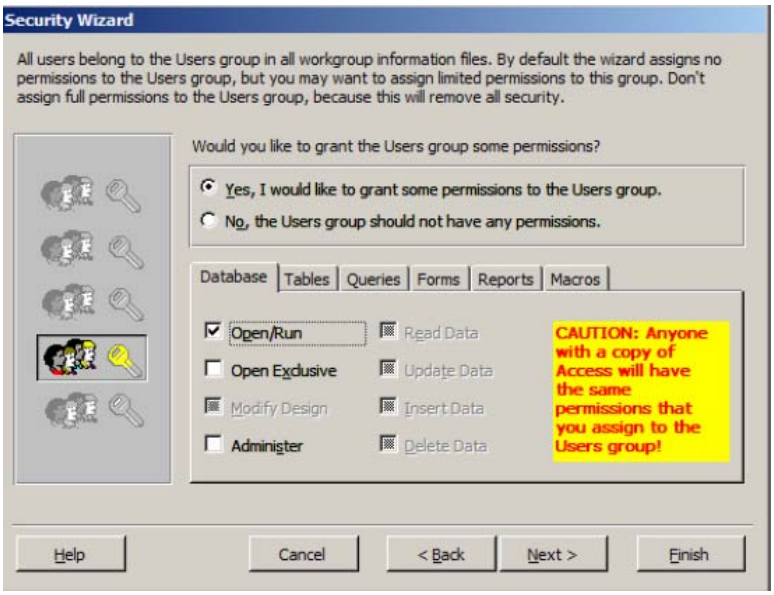

Figure 4: A denial all access model by default

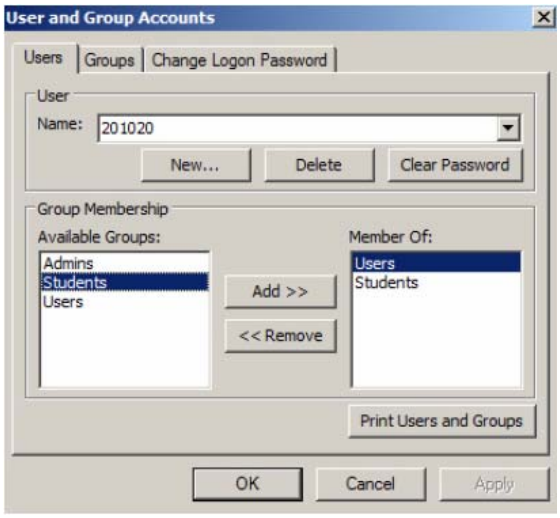

Figure 5: Assigning users to groups (roles) in Access

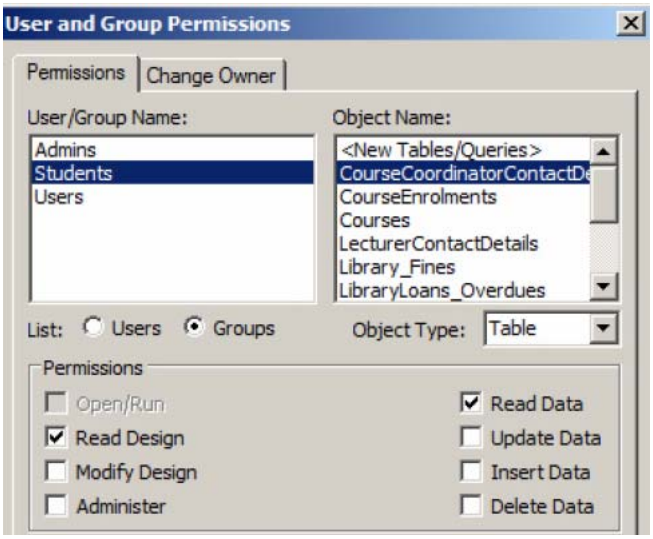

Figure 6: Assigning groups to objects with permissions

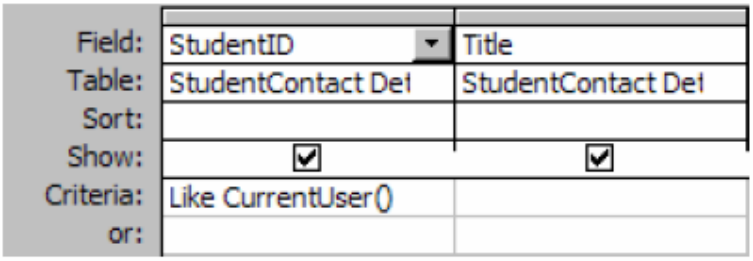

Figure 7: Linking logged in users to user specific data

BENEFITS OF ACCESS FOR SECURITY EMULATION

In the case of DB Sec the benefit of using Access as a database tool to emulate enterprise level security models and concepts is the flexibility provided in the teaching environment and final assessment. Most of the on-campus teaching labs in which DB Sec is taught have Microsoft Access installed, while students studying online can download and install a site licensed copy of Access should they not already have it at home. Access will run on almost any version of the Windows operating, again allowing students to work with system off-campus without need of specialised OS installs. If students run into trouble with their assessment work, it its extremely easy to bring in or even email their working files to their instructors for some feedback, without the needed for complex RDBMS setups and establishment of security accounts. This also applies for marking of student assessments, for as long as the students submit their Access database file with the associated .MDW security file, everything that is needed for marking the security implementation is in place.

Since the transition to using Microsoft Access as a security emulation system the delivery of DB Sec both in the class and online has become far less problematic from a teaching and assessment standpoint, while on the whole student reaction has been position due to the ease of use of the environment.

THE DOWN SIDE

While the use of Microsoft Access has made the overall delivery of DB Sec easier, it has come at the cost of true database security authenticity. Students still learn how field level security should work and can see how different levels of users should be able to see different data and perform different tasks. However, given the design roundabouts that are required in Access to achieve these goals, students could not directly apply the same skills to an enterprise RDBMS for the same results. While students are made aware of these drawbacks and are

on the whole satisfied with the learning outcomes, many still would like the opportunity to apply their security skills to a real RDBMS environment.

For these reasons it is expected that while Access is still the tool of choice for applying database security concepts in DB Sec, within the next year it might possibly be replaced by an RDBMS, most likely MySQL 5. Given the availability of MySQL as part of integrated packages, like XAMPP (www.apachefriends.de), there is now the opportunity to have an enterprise level RDBMS available for easy deployment to on-campus and off-campus machines. Coupled to a web-based user interface, such as phpMyAdmin (Figure 8), course developers and students alike have access to an easy-to-deploy, portable and authentic environment with which to apply their database security skills.

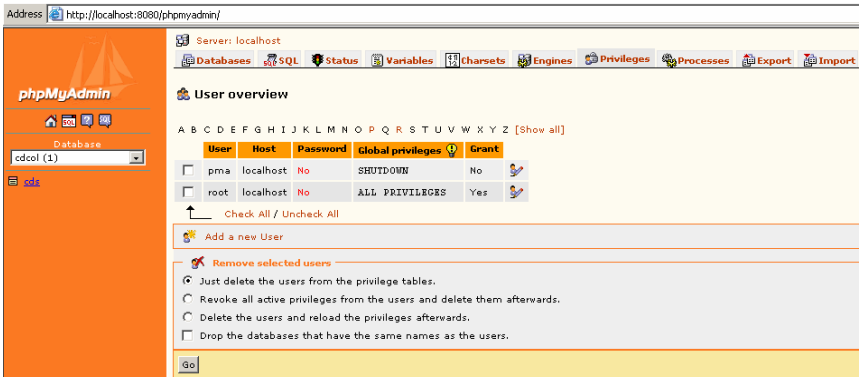

Figure 8: phpMyAdmin web-based interface onto MySQL 5

CONCLUSION

This paper has presented a possible alternate approach to teaching and applying database security concepts whereby the practical classroom teaching practice is not tied to large RDBMS environments that can be expensive, difficult to deploy in a managed environment and even harder to setup for students working off-campus. It has been shown that file based database tools such as Microsoft Access can simulate the application of database security concepts and methods, without actually being a secured database environment. While the benefits include easy of deployment and assessment, the simulation of security is just that, a simulation.

While moving to an authenticate RDBMS environment such as MySQL may provide greater real-world context for students studying DB Sec and the concepts contained within, it remains to be seen whether students will actually have fundamentally more database security knowledge than that which they are receiving through the Access solution currently in use.

REFERENCES

Bertino, E., & Sandhu, R. (2005). Database security - concepts, approaches, and challenges. *IEEE Transactions on Dependable and Secure Computing, 2*(1), 2-19.

Binto, G., & Anna, V. (2006). *A database security course on a shoestring*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

Brown, J. (2006, June 26-29). *Teaching Web Applications Development in a Fully Online Environment: Challenges, Approaches*

and Implementation. Paper presented at the 2006 International Conference on E-Learning, E-Business, Enterprise Information Systems, E-Government, & Outsourcing, Las Vegas, USA.

Bullers, W., Burd, S., & Seazzu, A. (2006). *Virtual machines - an idea whose time has returned: application to network, security, and database courses*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

Guimaraes, M. (2006). *New challenges in teaching database security*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

Guimaraes, M., Mattord, H., & Austin, R. (2004). *Incorporating security components into database courses*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

Haas, D. (1988). *Teaching database using a real DBMS: experience with INGRES*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

Srinivasan, S., & Anup, K. (2005). *Database security curriculum in InfoSec program*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

Udoh, E. (2006). *Teaching database in an integrated oracle environment*. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

COPYRIGHT