

From Database Concepts to Application: Use Problem-Based Learning and Oracle Development Tools to Facilitate Learning

Jennifer Kreie
jkreie@nmsu.edu
Accounting & Information Systems

Bruce A. Ernst
bernst@nmsu.edu
ICT Database Administration

New Mexico State University
Las Cruces, New Mexico 88003, United States

Abstract

Database concepts are an important part of the curriculum in both information systems and computer science majors and in business today relational databases are the foundation of most business applications. While relational database concepts are usually presented to students through lectures, textbooks, and online material, it is important to help students get beyond rote-based learning. This paper presents a teaching approach to help students move from conceptual knowledge to practical skills through problem-based learning that has students apply the database concepts by participating in the design, construction and use of a database. This paper describes the problem-based learning approach to teaching, outlines a series of tutorials that present database concepts and proposes a problem-based series of assignments that require students to apply concepts they have just learned. The problem-based learning approach should challenge students to understand database concepts and extend their knowledge. The hands-on tutorials as well as the problem-solving assignments given to students use free Oracle products: Oracle XE, Application Express (APEX) and Oracle SQL Developer/Data Modeler. The tutorials are publicly available on YouTube and instructor materials are available from the corresponding author. The appendices to this paper present information about the content of the tutorials, an example problem-based learning scenario, and instructions to help instructors install and use the Oracle products. The teaching approach presented here could be suitable for either an analysis and design course or a database course.

Keywords: problem-based learning, database management systems, data modeling, entity relationships diagram, ERD, relational database, normalization, teaching resources

1. INTRODUCTION

The topic of databases is an important part of the curriculum, whether it is a computer science program or information systems program in a college of business. It seems that students have difficulty grasping database concepts particularly with respect to database design (Murray and Guimaraes, 2009; Chilton, McHaney, and Chae, 2006) and this difficulty may contribute to the dropout rate (Kaufmann, Mense, Wahl, and Pucher, 2011). Since databases serve as the foundation of most business applications today it is important to explore ways to improve students' understanding learning of this topic. Typically, relational database concepts are presented to students through lectures, textbooks, and online material but this paper presents a problem-based learning (PBL) approach to help students move from conceptual knowledge to applied skills. Students first study database concepts and terminology then apply these by participating in the design, construction and use of a database.

This paper focuses on teaching relational database concepts and describes a series of tutorials that can be used in conjunction with covering database concepts and terminology. This paper proposes the coverage of concepts and use of the tutorials in parallel with a problem-based series of assignments that require students to delve deeper into the concepts and learn how these are applied and they do so by designing and building a database. The hands-on tutorials, as well as the problem-solving assignments given to students, use free Oracle products: Oracle Database XE, Application Express (APEX) and Oracle SQL Developer/Data Modeler. These products are briefly described within the body of the paper. The appendices to this paper provide additional information about the tutorials, an example problem-based learning scenario, and instructions to help instructors install and use the Oracle products.

2. PROBLEM-BASED LEARNING

Problem-based learning (PBL) is a teaching method most often associated with its use in medical schools where it has been used for years (Prince, 2004). The explanation of PBL varies somewhat in the literature but there are elements common to most descriptions of this teaching approach. Using PBL, teachers present

students with a problem at the beginning of instruction. The problem must be sufficiently complex that students cannot immediately solve it, and it must require students to go beyond rote-learning which is often sufficient for end-of-chapter problems in textbooks (Hansen, 2006). Solving the problem should require students to *actively* engage in research, analysis and exchange of ideas (Karantzas, Avery, Mcfarlane, Mussap, Tooley, Hazelwood, and Fitness, 2013). The teacher acts as a facilitator or tutor but not as someone who provides answers. Teachers can provide crucial assistance by helping students develop their problem-solving skills (Prince, 2004; Chaparro-Peláez et al., 2013). Though not always stated in the literature as a component, PBL is often applied in a collaborative learning setting in which students work in small groups. In a collaborative learning setting students within a group typically divide the research work then share among themselves what they have learned so that each student has the role of teacher for some issues related to the problem and solution (Prince, 2004; Hansen, 2006; Chaparro-Peláez et al, 2013).

The authors propose the use of PBL to help students learn relational database concepts. However, it is not suggested that students learn solely through discovery. Initially, some content is presented through lecture, handouts, readings, and/or videos but students are immediately challenged with a problem scenario and assignments that challenge students to think about the database concepts, gather additional information and develop a solution. The following plan for PBL can be done with students working individually or in small groups, as seen fit by the teacher.

The instruction and PBL assignments are in three phases: model, build, and use. The timeframe recommended is six weeks but this could vary. The addition of the "use" phase goes beyond what is often taught in a database class. However, this phase provides important feedback to students about how well database concepts are understood and applied. There is no need to expect students to quickly and completely master these concepts. In fact, trying to use a poorly designed database is probably the best way to understand the importance of concepts such as normalization. During the "use" phase and using Oracle APEX, students can quickly develop a prototype application and see the impact of their database design.

3. ORACLE DEVELOPER TOOLS

The PBL method presented in this paper does not require the use of a particular set of tools but this paper does describe the use of four free Oracle products (Oracle). Appendices C through E provide information to help instructors get started with these products. Here is a quick overview of the products.

Oracle XE is a free version of the Oracle database software. Individuals can install this as a desktop version but for instructional purposes it is necessary to install XE on a server. (See the license agreement for details about proper installation and use.) Installing XE on a server means all work done by students will be in one central location and installation issues on individual computers are avoided.

APEX (Application Express) is an application development tool that is automatically installed when the database software is installed. APEX runs through a browser and has several interfaces that provide tools for building and using a database. The Application Builder is one component of APEX that is a powerful tool for creating a web application that uses a database backend and that works in the most commonly-used web browsers.

SQL Developer is a GUI tool that provides connectivity to database accounts/schemas, usually for the purpose of developing and running SQL commands.

SQL Developer Data Modeler is a GUI tool used for creating logical data models which can then be translated into physical, relational models complete with SQL scripts to implement the construction of the database. The data modeler can be used as a stand-alone modeling tool and it can also be run within SQL Developer.

For the instructor, SQL Developer offers a very useful feature, a schematic diagram of the data dictionary from a schema, as shown in Figure 1. This makes it easy to review what students have implemented in their assignments for the database. For example, implementing a relationship shown in the model requires creating a constraint in the database but this is often overlooked when tables are built and, therefore, referential integrity is not enforced. The schematic diagram shows foreign key constraints so missing ones are evident.

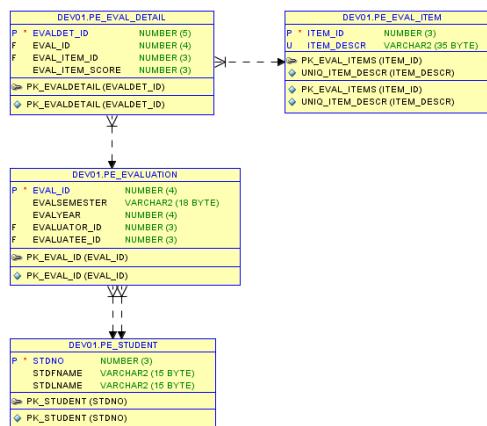


Figure 1

4. LEARN DATABASE CONCEPTS

Instruction – Modeling

The presentation of basic relational database concepts can be done through textbook material, online resources (ADBC at <http://adbc.kennesaw.edu>, for example), and material written by the teacher or a combination of these. The topics covered include:

- Data model components: entities, attributes and relationships.
- Diagramming conventions: Chen, crow's foot, IDEF1X, Bachman, Barker, etc.
- Cardinalities, types of relationships (one-to-one, one-to-many, and many-to-many) and associative entities.
- Logical models & physical models
- Relational databases
- Principles of normalization
- Primary, foreign keys, candidate keys, and surrogate keys
- Multivalued attributes

Appendix A has a scenario about a teacher that needs to keep track of teams, students and peer evaluations. The data model for this scenario, which is the basis of a series of video tutorials to learn how to use Oracle APEX, is used to present the above concepts to students. Two versions of the data model are given as shown in Figure 2. One has many-to-many relationships and the other has these relationships decomposed into one-to-many relations through the use of associative entities. Through discussion of these models, students start learning about database concepts and how they relate to an actual problem.

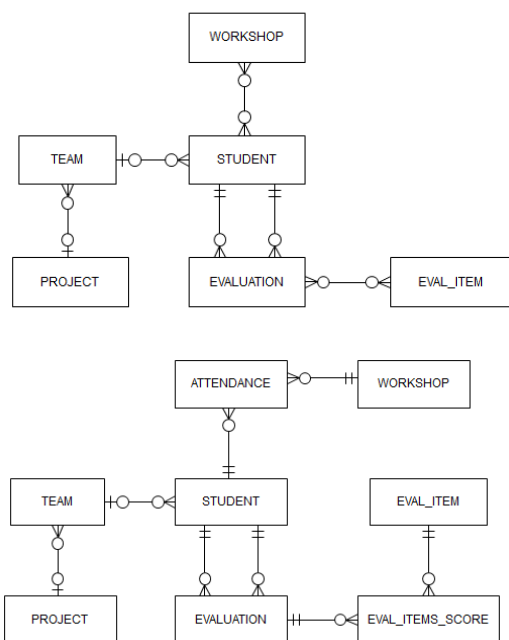


Figure 2

PBL – Model

Students are given the scenario in Appendix B called HAL (Home At Last) and are assigned the task of creating a logical data model for this scenario. SQL Data Modeler can be used to create this model.

The scenario raises some real-world design issues that may not have been covered yet with regards to data modeling. For instance:

- Do people with different roles, such as volunteer, staff or adopter, imply different entities or one entity?
- Is there information in the scenario that is not necessarily needed in the data model? How do we determine what to include and what not to?
- What purpose do business rules play in data modeling?

The teacher must strike a balance between guiding the students and letting them discover things for themselves. It might be beneficial to student learning in the long run if the teacher refrains from correcting some modeling errors. During the building or use phases, the impact of errors can offer a good opportunity to learn. However, the data model must be a reasonably good solution or implementing the design will create too many frustrations and failures.

5. APPLY CONCEPTS

Instruction – Building

During this phase and part one of the tutorial there are opportunities to teach additional database topics, such as:

- Database integrity based on primary key rules (unique and not null) and referential integrity
- Constraints (primary, foreign, and unique)
- Triggers
- Auto-generated values for primary keys (sequences in Oracle)
- Data quality issues and reference tables

There can be some repetition of modeling concepts when the transition from a logical data model to a physical one is demonstrated with the team tracking scenario. At the same time students should work through part one of the APEX tutorial which introduces students to many APEX features. Students learn quickly how to use the Object Browser tool in APEX to construct a database based on a physical database design. Scripts for the tutorials insert data in many of the tables which also helps students visualize how database tables work together.

PBL – Build

In parallel with instruction about physical versus logical models based on the team tracking scenario and the first part of the APEX tutorial, the assignment for students is to transform their logical model for the HAL canine rescue scenario, then to build the database in Oracle XE using APEX.

It is worth noting here that, although teachers may think the assignments based on the HAL scenario are too similar to the team tracking tutorials, there is anecdotal evidence that small differences in scenarios can be quite challenging for students as they synthesize concepts and apply them to a similar but different setting.

6. USE THE DATABASE

Instruction – Using

In tutorials two through six students learn that they can use APEX to quickly create a working prototype of a database application. Through the construction and use of a few forms and reports, students can see the impact of database design on user interfaces.

PBL – Use

Students create an application prototype for HAL based on the database they designed. Due to some of their decisions earlier in modeling and then building the database, some problems may be encountered. For example, if a student creates separate tables for people because of their different roles, they will encounter problems with some forms and reports. These are excellent opportunities for students to gain a better understanding and integration of database design concepts.

7. SUMMARY

This paper describes a problem-based learning approach to teaching database concepts in a database or analysis and design course. As concepts are covered assignments are given that have students apply and extend what they are studying. Either in an analysis and design course or a database class, the PBL approach can help students achieve database concepts.

8. REFERENCES

- Chaparro-Peláez, J. , Iglesias-Pradas, S., F. J. Pascual-Miguel and A. Hernandez-Garcia (2013) Factors affecting perceived learning of engineering students in problem based learning supported by business simulation. *Interactive learning environments*, 21(3), 244-262. (Journal 3 or more authors)
- Cilton, M.A. McHaney, R., and Chae. B. (2006) Data Modeling Education: The Changing Technology, *Journal of Information Systems Education*, 17(4), 17-20 (Journal 3 or more authors)
- Hansen, J. D. (2006) Using Problem-Based Learning in Accounting, *Journal of Education for Business*, March/April, 221-224. (Journal 1 Author)
- Karantzas, G. C., Avery, M. R. Mcfarlane, S., Mussap, A., Tooley, G., Hazelwood, Z., and Fitness, J. (2013) Enhancing critical analysis and problem-solving skills in undergraduate psychology: An evaluation of a collaborative learning and problem-based learning approach. *Australian Journal of Psychology*, 65, 38-45. (Journal 3 or more Authors)
- Kaufmann, C., Mense, A., Wahl, H., and Pucher, R. (2011) Reducing the drop-out rate of a technical oriented course by introducing problem based learning – a first concept. *Journal of Systemics, Cybernetics & Informatics*, 9(2), 51-55. (Journal 3 or more Authors)
- Oracle Corporation, Database Express (XE), <http://www.oracle.com/technetwork/product/s/express-edition/overview/index.html>, SQL Developer and Developer Data Modeler, <http://www.oracle.com/technetwork/product/s/express-edition/overview/index.html>, (Web pages).
- Prince, M. (2004) Does active learning work? A review of the research. *Journal of Engineering Education*, 93, 223-231. (Journal 1 Author)
- Murray, M. and Guimaraes, M. (2009) Animated Courseware Support for Teaching Database Design, *Issues in Informing Science and Information Technology*, 6, 201-211 (Journal 2 Authors)

Appendix A

Scenario for Database and Oracle Tutorials

Team Tracking Scenario

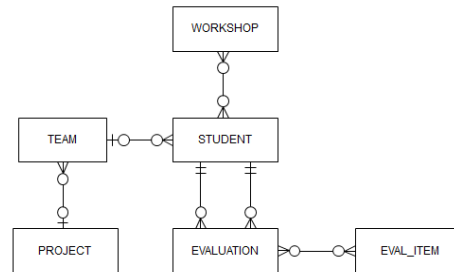
Judy Westfall is a teacher at a university in the southwest United States. In two classes that she teaches regularly, she has students work in teams on major class projects. For now, Dr. Westfall keeps records of the teams and the students on each team in a spreadsheet with multiple worksheets. When teams are formed each team decides on a team name and each team may be assigned a project at that time.

It is important to keep track of peer evaluations, which are evaluations done by team members at the end of the project. Each team member must evaluate himself/herself and the other team members. Evaluations are ratings on a scale from 0 to 10 (best rating) for things such as, "on time to meetings," "listens to others," "contributes ideas," etc. Currently, students complete a paper evaluation form and hand it in to Dr. Westfall, who then enters the scores in a spreadsheet and calculates the average rating for each student in each team for each evaluation item. The individual evaluations done by each team member are kept confidential to help encourage honest evaluations. After each project is graded, each student gets a summary evaluation report that shows his/her average evaluation score for each evaluation item. For example, Susan Anthony might get a report that shows her average ratings for "on time to meetings," "listens to others," "contributes ideas," were 9.7, 8.9, 9.0, respectively. This report is meant to provide feedback to team members about what their teammates thought of their work on the team, and the scores are used by Dr. Westfall in calculating the individual score for each team member for the project. Each student also receives a summary of the average evaluation ratings for the team.

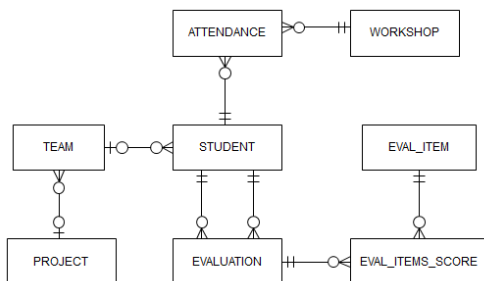
To help students think about and develop teamwork skills, there are several workshops available for teamwork skills, such as "resolving conflict," "professional behavior," etc. Some workshops require participation in a one or two-hour face-to-face session, or they may be self-paced sessions requiring reading and exercises. These workshops are not required but students can earn extra points for doing them, and the data from past semesters indicates that students usually receive higher marks on their evaluations if they complete two or more workshops.

Dr. Westfall needs a better way to track teams, team members, evaluations, workshops and workshop attendance.

Data model with many-to-many relationships.



Model with many-to-many relationships decomposed.



Teams Tracking: A tutorial for Oracle APEX using an Oracle database

OUTLINE OF TUTORIALS AND VIDEOS

Part 1 - (Videos 1-7) *Some parts of video #1 only apply to the instructor. Students do not need to install XE and create their own APEX workspace.*

- Overview of Oracle APEX tools
 - Application Builder
 - SQL Workshop
 - Object Browser
 - SQL Commands
 - SQL Scripts
 - APEX workspace
- Workspace administrator
- Workspace developer
- End user
- Create a developer account using the APEX Administration tool
- Create some tables using SQL Scripts
- Create triggers.
- Add data (insert into ...) in the tables
- How to create a table in the Object Browser
- View sequences in the Object Browser
- Define a constraint for a table
- Import data to a table
- Create a Lookup Table

Part 2 - (Videos 8-9)

- APEX application – Pages, headers, footers, regions and items
- Oracle schema and APEX parse as schema
- Create an application
- Page layout & tabs
- Authentication method
- Public pages
- Design theme
- Create a simple, interactive APEX form with report

Part 3 - (Videos 10-11)

- Set defaults for table and column attributes using the Object Browser
- Edit a tab and change navigation to a page
- Shared Components
- Set UI defaults for all the tables
- Plan what reports and forms are needed
- Form with report (one table). Six of these are created
- Master-Detail form with report (two tables).

Part 4 - (Videos 12-13)

- Modify forms generated by APEX
- List of Values (shared component)
 - APEX can help write the SQL query
- Add LOVs to a form
- Show a "hidden" column

Part 5 - (Videos 14-15)

- Master-Detail form with report (two tables)
- Create a region in a page and add buttons to a region

Note: Depending on time constraints, it is possible to omit parts 6 and 7.

Part 6 - (Videos 16-19)

- Create a LOV with more than one field of data in the display column
- SQL Workshop – Creating views
- Create an APEX classic report with download option
- Create an APEX interactive report based on a view

Part 7 - (Video 20)

- Create a master-detail form and add another detail section

List and hyperlinks to videos on YouTube for this tutorial.

Oracle APEX Tutorial (01 of 20) - [Using Oracle XE](#)

Oracle APEX Tutorial (02 of 20) - [Overview of APEX](#)

Oracle APEX Tutorial (03 of 20) - [Run scripts to create tables & sequences](#)

Oracle APEX Tutorial (04 of 20) - [Create triggers and insert data using SQL Workshop](#)

Oracle APEX Tutorial (05 of 20) - [Add a constraint to a field in a table](#)

Oracle APEX Tutorial (06 of 20) - [Import data using tools in SQL Workshop](#)

Oracle APEX Tutorial (07 of 20) - [Lookup Table Wizard creates a reference table](#)

Oracle APEX Tutorial (08 of 20) - [Create an application](#)

Oracle APEX Tutorial (09 of 20) - [Create a report and simple form](#)

Oracle APEX Tutorial (10 of 20) - [Improve forms and reports: modify user interface \(UI\) defaults](#)

Oracle APEX Tutorial (11 of 20) - [Plan what forms and reports are needed](#)

Oracle APEX Tutorial (13 of 20) - [Show a hidden field \(page item\) in a form](#)

Oracle APEX Tutorial (14 of 20) - [Create a master-detail form](#)

Oracle APEX Tutorial (15 of 20) - [Add a region to a page and buttons to a region](#)

- *Tutorials after 15 might be considered optional depending on time constraints and the scope of topics being taught.*

Oracle APEX Tutorial (16 of 20) - [LOV \(list of values\) with concatenated columns](#)

Oracle APEX Tutorial (17 of 20) - [Create views \(stored queries\)](#)

Oracle APEX Tutorial (18 of 20) - [APEX classic report & export to CSV file](#)

Oracle APEX Tutorial (19 of 20) - [APEX interactive report](#)

Oracle APEX Tutorial (19b of 20) - [More about interactive report features](#)

Oracle APEX Tutorial (20 of 20) - [Master form with two detail report sections](#)

Appendix B

Example Scenario for Problem-Based Learning

Home At Last (HAL) – Canine rescue scenario

David Hernandez and Jessica Akers became friends several years ago when they met at a dog training class and found they both loved to work with Blue Heelers, also known as Australian Cattle Dogs. Five years ago a friend of theirs offered to donate five acres with some storage buildings if they used it to establish a non-profit rescue facility. The organization is now called HAL (Home At Last).

Thanks to contributions and successful fund-raising, HAL now has one full-time and three part-time employees. Volunteers also work at the facility to care for the dogs—feeding, grooming, training and exercising the dogs.

Although David and Jessica originally focused on Blue Heelers they have begun accepting other types of dogs, as well. HAL receives dogs from owners who can no longer care for them and from people who find abandoned dogs. It is fortunate that four local veterinarians provide medical services at a reduced rate to HAL.

David, Jessica, the staff and volunteers provide training to help dogs overcome behavioral problems and to help puppies learn manners, which makes it easier find homes for them. Part of the process for placing a dog in a home is a required six-class course over three weeks during which the prospective owner and dog work together. This gives the person and the dog time to get to know each other and determine if the dog will be a good match for the prospective owner's home.

Everyone at HAL agrees that the way they currently keep track of information about the dogs and people associated with HAL needs improvement. HAL needs to track information about dogs, staff and volunteers, dog donators, and dog adopters.

Problem statement

You will design and build a database for HAL. Once the database is in place you will build an application that allows the HAL staff to maintain information about dogs and the people associated with HAL—staff, volunteers, people who bring dogs to HAL and people who adopt dogs from HAL. The database should handle the data needs described below, and the application should give HAL the ability to maintain this data.

Things HAL needs to know about a dog: breed, estimated birth date (so age can be estimated), name, physical description, date received at HAL, comments about personality, sex, spayed or neutered, passed obedience training (yes/no), and status (this could change over time).

Things HAL needs to know about people: name, phone numbers, email address, mailing address, and street address.

Things HAL needs to know when receiving a dog: name of the person who brought the dog to HAL, explanation about why the dog was taken to HAL, the date HAL received the dog, and any comments about the reason the dog is now at HAL.

Things HAL needs to know about a dog adoption: name of the person who adopts the dog, name of the person at HAL (staff or volunteer) who processes the adoption, date of the adoption, and any comments about the adoption.

Appendix C

Installing Oracle XE and Upgrading APEX

These instructions are written based on the assumption that the reader has some knowledge of SQL and the operating system (Windows or Linux) in which the software is installed.

Oracle XE

Currently, the link to this product is: <http://www.oracle.com/technetwork/products/express-edition/overview/index.html>. If the links no longer works, do a web search for "Oracle XE" (Oracle Express Edition) to find the download link at Oracle's web site.

In order to download XE and other Oracle products, you will have to create a free Oracle TechNet (Technology Network) account, if you don't already have one. You will also have to agree to the license agreement.

At the same site as the download link for XE, you will also see Oracle's documentation for this product. Currently, the link to installation instructions on a Windows operating systems is at: http://docs.oracle.com/cd/E17781_01/install.112/e18803/toc.htm#CIHHJEHF

The installation is simple and fairly quick. You can accept the installation defaults. Be sure to record the passwords you set for the database administration accounts, SYS and SYSTEM.

Application Express (APEX)

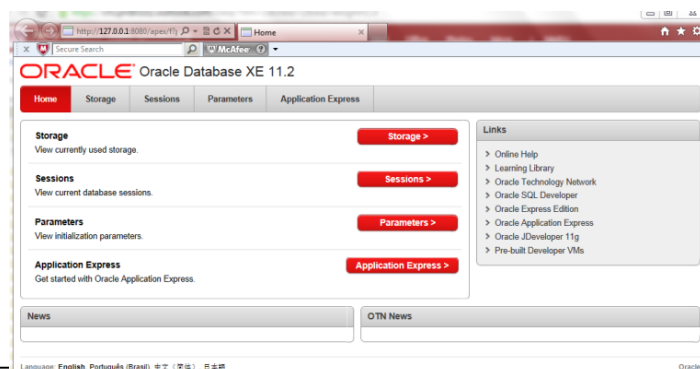
When XE is installed APEX is also installed. However, the version installed is often two or three versions older than the most current version. You can work with the installed version or upgrade APEX. To get the most current version, do a web search for "Oracle APEX download" and download the software. Current link to download APEX: <http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html>

- Installation guide: http://docs.oracle.com/cd/E37097_01/doc/install.42/e35123/overview.htm#HTMIG363
- Since this installation is for educational purposes, it is recommended that you use the embedded PL/SQL Gateway installation because it is simple to do and the security issues about running the HTTP listener and database listener on the same machine aren't as much of a concern.

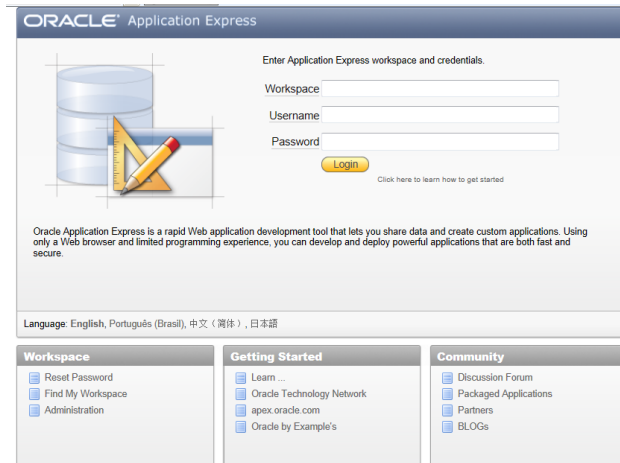
Installation Completed

Oracle XE: The XE interface is shown here. The URL for XE on the local computer is: <http://127.0.0.1:8080/apex/f?p=4950>

There is a video available (see Appendix A) that explains how to get started with APEX using this interface but, as an instructor that uses APEX in class, you probably won't use this much, if at all. The APEX application (see below) and SQL Developer are much more useful tools.



APEX: The URL to go straight to APEX when Oracle XE is installed on the local computer:
<http://127.0.0.1:8080/apex/apex>



Note 1: Once XE and APEX are installed you can use the video "Oracle APEX Tutorial (01 of 20) - [Using Oracle XE](#)" to see how to get started using APEX. The appearance of XE and APEX in your installation and the ones shown in the video may vary somewhat because of differences in the version shown and your version of the software.

Note 2: The authors generally don't use the XE graphical interface when setting up accounts to use for the tutorials and assignments. SQL scripts have been created that do a lot of this set-up work and the scripts are run through SQL Developer (see other appendix for this software).

Note 3: When XE is installed the operating system settings are changed so that the database is automatically started whenever the computer is turned on. In Windows there is a menu option to start/stop the database (Start Menu ... All Programs ... Oracle Database Express Edition ...). You may wish to change automatic startup by changing the Windows services setting to "manual" instead of "automatic." For steps on how to do this, do a web search for how to change services settings in Windows.

Appendix D

Installing Oracle SQL Developer and Data Modeler

Download and install

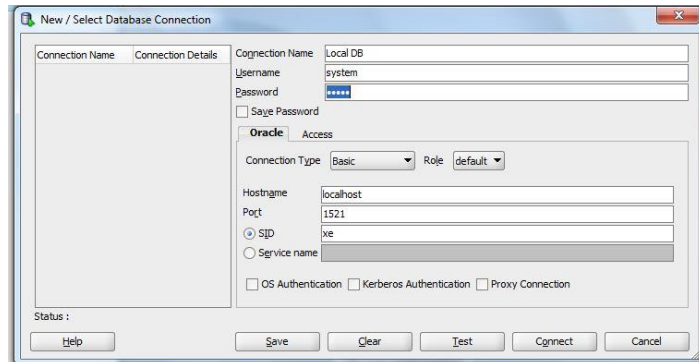
Currently, the link to this product is: <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>. If the link doesn't work, do a web search for "Oracle SQL Developer" to locate the software and download it. You should also see SQL Developer Data Modeler listed on the overview page along the left side, so you can download that as well. It is recommended that you create a folder for each download and put these in a place where you keep track of software.

Each of these downloads puts a zipped file in the location you specified during the download. Unzip each file in the folder you created for it. After extracting the files there will be an executable file that launches the program (sqldeveloper.exe or datamodeler.exe). Use these to open each program. (Create shortcuts to the desktop, if you wish.)

SQL Developer

SQL Developer is a GUI interface for connecting to a database account/schema. When you open SQL Developer you need to create a connection to the database.

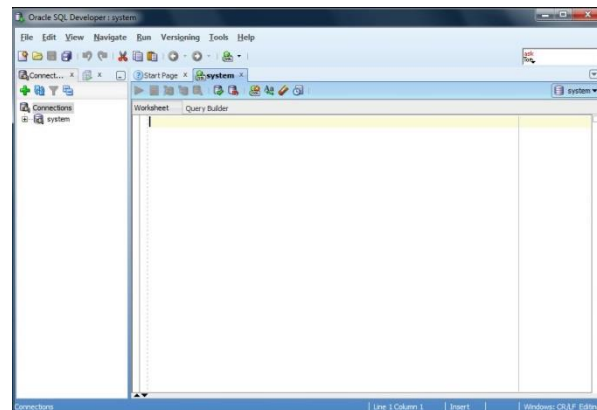
The connection name (top row) is simply a text label that appears in SQL Developer after the connection is configured. The user name and password are based on which account you want to use.



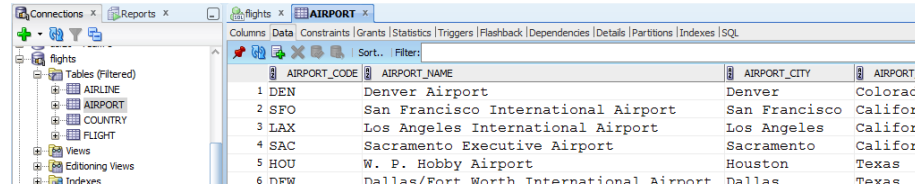
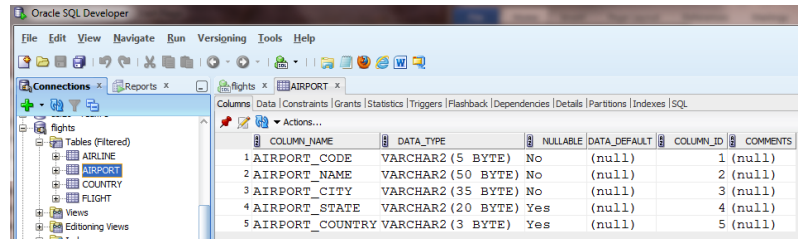
Note: Don't check "Save Password" if this computer is used by others than yourself.

The default connection settings (connection type, hostname, port, SID) are for a locally-installed Oracle XE so you don't need to alter these. Once you configure a connection you will see the connection name listed on the left side of the screen any time you open SQL Developer. You can double-click the connection name to open that connection. If the password is not stored, you will be prompted to enter it.

Here is an example of the SQL Developer worksheet that appears when a connection is established. Although SQL commands can be entered and executed in the right-side pane, there are also graphical views of many database objects.



Expanding the connection in the left-side pane provides a list of items, some of which can be viewed in the right-side pane. These two screen shots show a table selected on the left and the definitions of columns (1st) and the data in the table (2nd).

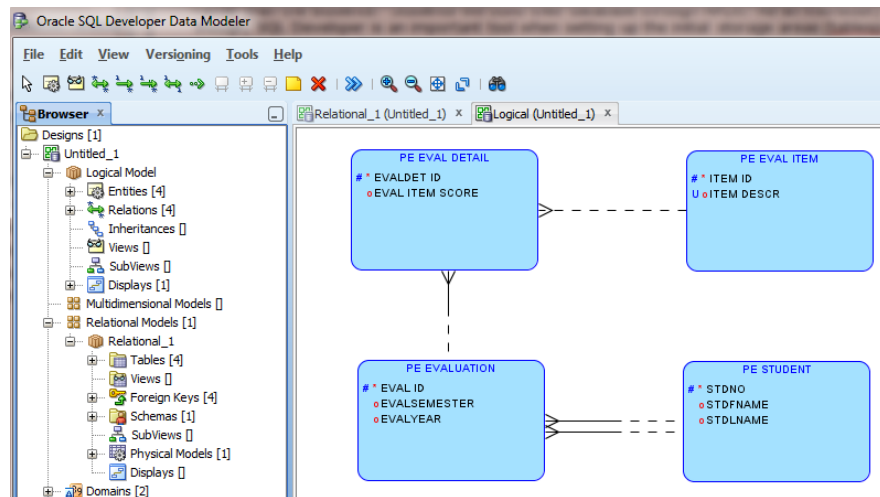


For the teaching method presented in this paper, SQL Developer is primarily used by the teacher rather than the students. Students will build their database through APEX. As an instructor, though, SQL Developer is an important tool when setting up the initial storage areas (tablespaces) and schema accounts for use in APEX. Note: Example SQL scripts can be obtained from the corresponding author of this paper.

Data Modeler

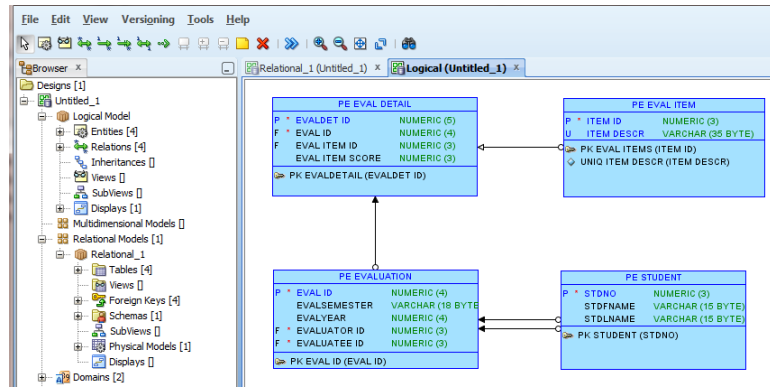
The Data Modeler is used to create logical data models and physical relational database models. If you draw a logical model, Data Modeler can convert this to a relational model.

The notation in the data model is called the Barker notation (Barker joined Oracle in the early 1980s). It is somewhat different than the more commonly seen crow's feet notation but it is fairly easy to make the transition between the notation variations.

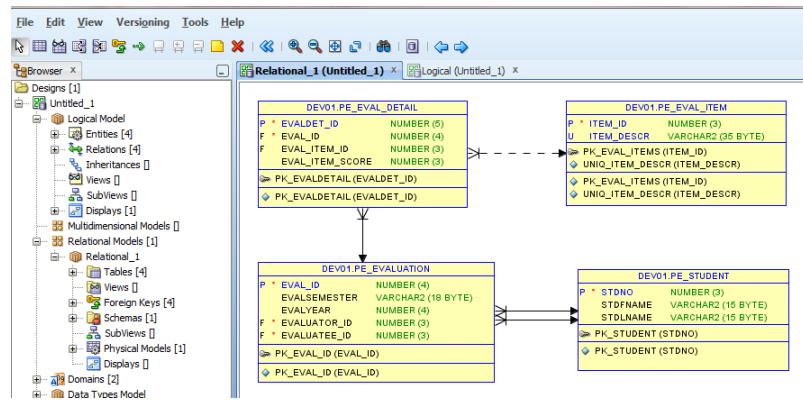


The difference in notation between Barker and crow's feet is seen in the minimum cardinality. The relationship line itself depicts the minimum. A solid line means "mandatory" (must have) and a dashed line means "optional" (may have). When reading the relationship of EVALUATION to EVAL DETAIL, the notation shown reads as: An evaluation may (dashed) have an eval detail. An eval detail must (solid) be part of an evaluation. The maximum cardinality is shown as "many" with the crow's foot and as "one" when there is a single line (no crow's foot).

There is one alternative notation called the Bachman notation, which shown to the right. A right-click in the diagram area allows you to switch between the two notations.



The relational model (the physical database model) is also shown here. The Data Modeler can convert a logical model to the physical model and vice versa.



Appendix E

Extract a Data Dictionary and Verify Database Construction

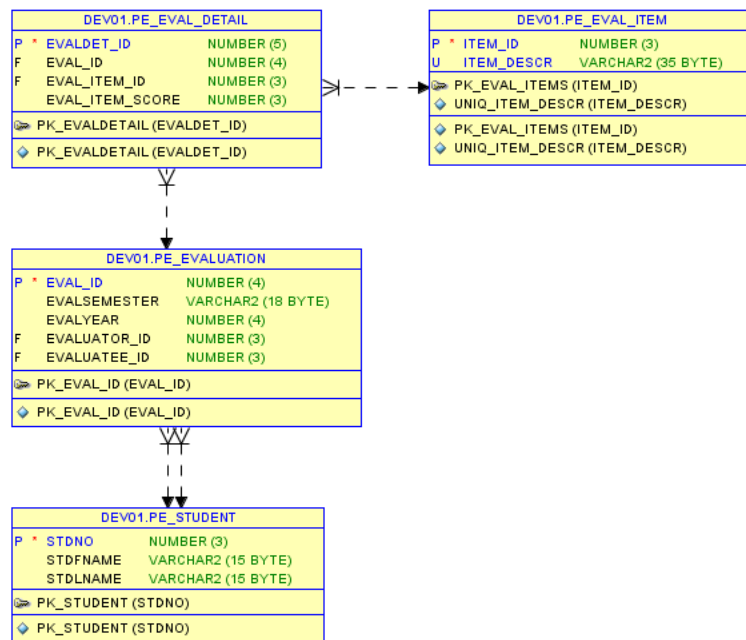
With either SQL Developer or Data Modeler you can extract the meta data about tables and constraints in a schema. This is very useful in verifying how students constructed their database. For example, it is common for students to build the tables but forget to build a foreign key constraint that enforces the relationship shown in the data model. In APEX you can see detailed meta data about each table but being able to see a relational model based on the meta data makes it easier to review how the database has been built.

How to do get this graphical data dictionary:

- From SQL Developer select **File** from the menu ... **Data Modeler** from the list ... **Import ... Data Dictionary**.
- Select the connection to the database you will use from the list and click the **Next** button.
- Select the schema/database.
- Select the tables to import and the graphical relational model will be generated.

In the example shown here we see:

- A rectangle for each table with name shown as <schema name>.<table name> (example: dev01.evaluation).
- Within a table there is a list of column names and their data types.
- The letter "P" to the left of the primary key column(s).
- The letter "F" to the left of columns with a foreign key constraint.
- The names of the constraints, such as for primary keys and foreign keys, are shown in the lower section of the rectangle. Other constraints, for example a uniqueness constraint, would also show up in the lower section of the rectangle.

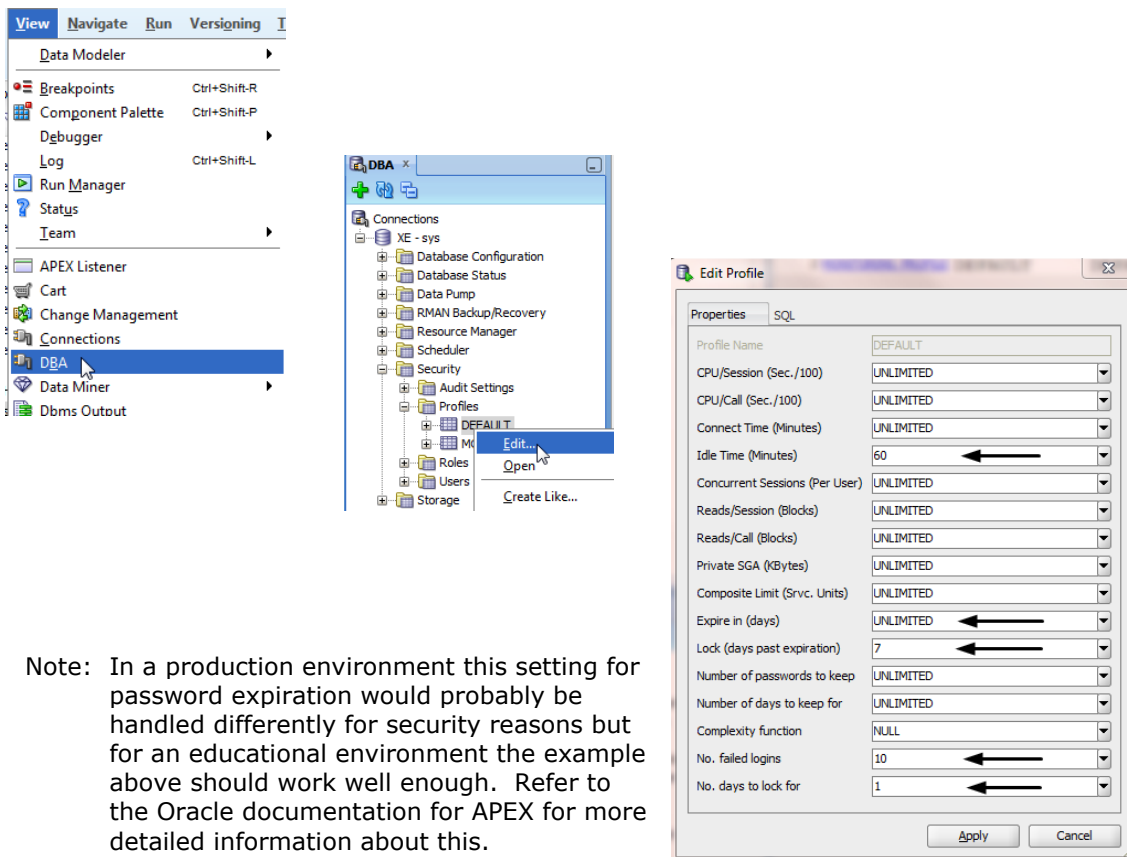


Appendix F DBA Tips for Instructors

Avoid password expiration

By default, the accounts created in APEX may be set to expire a password after 180 days. You might want to change this. The duration of a password is set in a profile. (A profile manages the security and connection properties for multiple accounts.)

1. Use SQL Developer to log in to the Oracle database as SYSTEM.
2. You can alter the default profile which is actually named "DEFAULT." (New users are assigned to the "DEFAULT" profile.) You can do this from the DBA pane. View this pane by selecting it from the View option in the menu bar. To extend the time before a password expires, change the field "Expire in (days)." You can turn off password expiration by setting it to "UNLIMITED." The example below shows changes to some other settings but these are only suggestions.



Note: In a production environment this setting for password expiration would probably be handled differently for security reasons but for an educational environment the example above should work well enough. Refer to the Oracle documentation for APEX for more detailed information about this.

Manage the database schemas outside of APEX

Create the schemas for students/teams yourself using SQL commands rather than having APEX create them because APEX will give lengthy arbitrary names.

1. Use SQL Developer to log in to the Oracle database as SYSTEM.

-
2. In the SQL Worksheet window run the following commands to create a role and assign several system privileges to that role. The system privileges listed below are the same ones the APEX "Create Workspace" wizard assigns.

```
CREATE ROLE apex_user_schema;  
GRANT CREATE JOB TO APEX_user_schema;  
GRANT CREATE DIMENSION TO APEX_user_schema;  
GRANT CREATE INDEXTYPE TO APEX_user_schema;  
GRANT CREATE OPERATOR TO APEX_user_schema;  
GRANT CREATE TYPE TO APEX_user_schema;  
GRANT CREATE MATERIALIZED VIEW TO APEX_user_schema;  
GRANT CREATE TRIGGER TO APEX_user_schema;  
GRANT CREATE PROCEDURE TO APEX_user_schema;  
GRANT CREATE SEQUENCE TO APEX_user_schema;  
GRANT CREATE VIEW TO APEX_user_schema;  
GRANT CREATE SYNONYM TO APEX_user_schema;  
GRANT CREATE CLUSTER TO APEX_user_schema;  
GRANT CREATE TABLE TO APEX_user_schema;  
GRANT CREATE SESSION TO APEX_user_schema;
```

3. Create a tablespace that all the student schemas will use. Having this tablespace will separate Oracle system data from student data and help you see the storage demands APEX workspaces have while they are being used in your classes. (A tablespace is a logical name for grouping tables and indices with common properties, such as all tables and indices in an application. It consists of one or more physical storage files. Storage capacity can be added by adding more storage files.)

In the SQL Worksheet pane run the "create tablespace" command based on the examples below. You will probably have to alter the path to the datafile based on how Oracle is installed on your server and whether the operating system is Windows or Linux.

Linux example:

```
create tablespace apex_student_schemas  
datafile '/opt/app/oracle/oradata/orcl/apex_student_schemas_01.dbf'  
size 200M;
```

Windows example:

```
create tablespace apex_student_schemas  
datafile 'c:\oracle\app\oracle\oradata\xe\apex_student_schemas_01.dbf'  
size 200M;
```

Note: The file size of 200M should be altered based on how many accounts you plan to use. Plan for at least 10M for each account for the tutorial and assignments. So, 20 accounts (students) means at least 200M should be allocated to the data storage file. However, additional data files can be added, if more storage is needed later.

4. Create an account for each schema that will be assigned to a workspace in APEX. Generally speaking, you will probably want an account for each student/team. Below is example SQL code for creating one account and assigning a profile, role and tablespace.

```
create user isys_01          -- ISYS_01 is the account name. Alter this as you wish.  
IDENTIFIED BY isys_01     -- This sets the password to "isys_01" and it is case sensitive.  
DEFAULT TABLESPACE apex_student_schemas  
TEMPORARY TABLESPACE temp  
PROFILE default;
```

```
GRANT APEX_USER_SCHEMA to isys_01; -- This grants all the privileges assigned in #2.  
ALTER USER isys_01 QUOTA 10M ON apex_student_schemas;
```

5. At the end of the semester, you can remove the student schema and all objects owned by the schema with the following command:

```
drop user isys_01 cascade;
```

To prepare for the next semester, the tablespace and role already exist, so you only need to create the student schema, as shown in step 4 above.

Run table statistics to help Oracle run efficiently

The Oracle optimizer uses objects statistics, such as the number of and average size of rows in table, to determine the most efficient method to access the tables in a SQL statement. For example, the optimizer will determine which indices to use, if any. As table data are changed, stale statistics can cause Oracle to select non-optimum table access methods. To bring the statistics back up to date, run the following PL/SQL block (BAE: use command instead?) in the SQL Worksheet pane in SQL Developer:

```
begin DBMS_STATS.GATHER_DATABASE_STATS (100); end;
```

Because this command scans all user tables in the database, it can take several minutes to execute.

It is especially important to gather these statistics after the schemas are initially populated with data, because new objects have no statistics.