

## CIS 315 - Database Design and Implementation - Spring 2002 Project Handout #1

This **individual** project will include the design and implementation of a database using the Oracle RDBMS on redwood. It must be completed, and all required components turned in, by **Friday, May 10th (by 4:30 p.m.)**. (Remember, as noted in the course syllabus:

**"The final project, like all assignments and project milestones, must be turned in on-time --- it will NOT be accepted late.** However, if the final project is turned in **before 4:30 p.m.** on the day **before** the due date, it will be considered "early", and will receive a bonus: 5 additional points (5/100) added to the project grade."

Note that this "bonus" does **not** apply to partial final projects.)

However, even though May 10th is the final due date, you will **also** be required to turn in parts of the project along the way, as will be specified. The first piece to be turned in will be a proposal describing the scenario for which you will be developing a database. (You can consider this sort of the **planning phase** of your project, if you will --- you will be defining the problem you wish to solve, establishing the scope of your solution, and assessing the feasibility of your proposal.)

Some suggestions for helping you to decide on your project proposal/scenario will be listed on the course web page. I will look at your proposals, attempt to let you know if they are too broad or too narrow, and then either approve or disapprove them. **Your proposal, which must be typed and included within a project notebook following a very particular format, is due on Wednesday, February 6th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture).** I should let you know a week later if your proposal is approved or not. If it is not approved, you will need to modify it until it is acceptable (but you are still bound by the **same** milestone deadlines as everyone else).

The next thing you will turn in will be the project notebook that now includes a **database model** for your project, which may be in the form of **either** entity-relationship diagrams **or** semantic-object diagrams. Your database model will be due on **Wednesday, March 6th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture).**

Following that, you will use the database model you come up with to come up with a **database design**. The project notebook that includes this will be due on **Wednesday, April 10th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture).**

On **Wednesday, April 24th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture)**, you will turn in your project notebook that includes evidence that you have successfully created the tables in your design, and populated them with sample data.

That will leave about two weeks until the final version of your project notebook, now including your implemented, completed database and other final project materials, will be due (by **May 10th, by 4:30 p.m.**).

You may, of course, work faster than the various deadlines imply! These are just the dates that the specified items must be turned in.

Summarizing important dates (all except the final project with due times of 1:10 p.m. if **any** of that class lecture is missed, and of 4:30 p.m. if you attend **ALL** of that lecture):

<b>Wednesday, February 6th</b>	- database proposal due
<b>Wednesday, March 6th</b>	- database model due
<b>Wednesday, April 10th</b>	- database design due
<b>Wednesday, April 24th</b>	- database population due
<b>Wednesday, May 10th (by 4:30)</b>	- final project due

### Grading Comments

Your final project should be robust enough to **demonstrate** as a **prototype** --- do not attempt to implement a "production quality" system! (And, if you cannot implement everything you hoped to, implement as much as you can, aiming for an **interesting, demonstrable prototype**.) A project that meets all of the minimal requirements mentioned in this handout, and that has met all of the minimal requirements all down the line (from the proposal on down), and that does so well, would receive a grade of **90**. The other **10** points I will reward based on merit --- did you come up with a particularly interesting, original proposal? is a database particularly well-designed and implemented? Is something about a project above-and-beyond the minimal requirements, or exemplary? Were useful extra features included, or was some major aspect particularly well-done? Did something about a project just stand out, or make a strong impression?

Please note, also, that you will receive grading comments for each milestone. These comments may include corrections or additions that must be made to your project. **If these are not done, each subsequent milestone's grade may be affected.**

**Milestone grading breakdowns:**

- \* the overall project grade makes up **35%** of the final course grade, as mentioned in the syllabus.
  
- \* this overall project grade is made up of:
  - \* project proposal: up to **7.5** pts
  - \* database model: up to **15** pts
  - \* database design: up to **15** pts
  - \* database population: up to **7.5** pts
  - \* final version of project: up to **45** pts baseline, plus up to **10** points for doing more than the minimum (as described above)
  
- \* (the overall project grade is then the sum of the above, and this overall project grade is multiplied by 0.35 to compute the project portion of your final course grade.)

(continue to next page, please)

## Detailed specifications of what is required for each milestone:

### In general:

- \* new or revised items should always be placed at the **beginning** of a section, in **front** of the already-existing items. Then, whenever I look in a section, your newest work in that section will be the **first** thing that I see.
- \* **every** item included in **every** section should include on it a clear **Last modified:** or **Last modified/printed:** date. This is an important habit to get into for items that may change over the life of a project --- it makes it far easier for someone to tell which is the latest version of something.
- \* note that not following the specifications will affect your grade, both on the individual milestones and on the final version of the project.

### Database minimum structural requirements:

Because an important topic in this course is database design, your project is required to meet certain criteria to help ensure that it is at least a little bit interesting in a structural sense. **Final projects that do not meet these minimum criteria will be severely penalized.**

The database model for your database must include:

- \* at least **5 distinct, significant** (classes of) **entities/objects**
  - \* note that this is NOT counting tables --- it is counting entities/objects. As you will find out, each entity in a model will be converted into **one or more tables** in the corresponding database design.
  - \* a superclass entity, and all of the subclass entities of that superclass, count as a **single** entity toward the five-entity requirement. Be careful!
- \* at least **4 relationships**
  - \* at least **one** of these relationships must be a legitimate **1:N** relationship.
  - \* at least **one** of these relationships must be a legitimate **M:N** relationship.

**What to turn in for database proposal on February 6th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture):**

Inside a **3-ring binder** with **10 dividers** and **one pocket**, you should have the following (!!! is placed before those sections whose contents will be **particularly significant** in grading this milestone):

\* (please note: by "**one pocket**", I do **not** mean a side pocket in the binder! I mean a **separate** pocket-page, that can be placed **within** one of the sections (not as part of the section divider, either). It will eventually hold a diskette within one of the sections in the final version of the project. It can be home-made, but I want it in here now.)

\* your **last name**, and '**Spring 2002**', must appear neatly on **BOTH** the **front** of the binder **and** on its **spine**.

**!!!\*** the **first** divider should be neatly labeled, on its **tab**, as **proposal**

\* after that divider should be your **typed** database proposal, including your name and the **date**, in which you:

- \* **clearly** define and describe the **scenario** of the problem you wish to solve (at least one full paragraph),
- \* **clearly** define and describe the major problem(s) that the users in the given scenario would like solved (at least one separate, full paragraph),
- \* include a **list** of at least **five** significantly-different questions that the users of your proposed system would like to be able to have answered using your completed database (separate from the above paragraphs),
- \* include a **list** of at least **three** significantly-different kinds of reports that the users of your proposed system would like to be able to generate using your completed database (separate from the above paragraphs) (note that a report is different from a question --- it includes more than one row or column of information, for example. See Kroenke and the final project milestone description for more description of what reports should be...),
- \* discuss the **feasibility** of your proposal (at least one full paragraph),
- \* specify any **assumptions** you need to make (if appropriate).

\* (please note --- you should **not** be mentioning tables or relations at this point **at all**.)

**!!!\*** the **second** divider should be neatly labeled, on its tab, as **business rules**

\* after that divider should be your **typed** introductory set of **business rules**, clearly marked with a **Last modified** date.

- \* business rules are operational, day-to-day "rules of thumb" in your project scenario --- they should be user-oriented, and **not** database-oriented. "A student must sign up for at least 1 class, and no more than 8", for example.
- \* (you should be able to come up with at least 8-10 such rules, even at this early stage.)

**Please note:** I will be looking at your proposals carefully and will attempt to deduce whether they are **likely** to result in a database that meets the minimum structural requirements mentioned earlier (at least 5 distinct, significant entities, and at least 4 relationships, at least one of which is 1:N, at least one of which is M:N). You will **not** be addressing these requirements specifically in your proposal, however, because that is **not appropriate** for this stage. You will make sure that they are met in your database model in the next milestone.

I will not accept proposals that seem likely to result in a database that does not meet these minimum requirements. Proposals that are not accepted must be modified so that they are acceptable, and please note that subsequent deadlines will **not** be changed as a result. Proposals also may be conditionally accepted --- this means that I think they could meet the requirements, but it depends on how you proceed. Whether proposals are accepted or not, however, you **must** meet these minimum requirements as you proceed.

I will try to note if a project looks **too** large/ambitious, but you should also be warned that such judgements are very tricky to make.

**What to turn in for database model on March 6th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture):**

Your binder, which now contains (!!! is placed before those sections whose contents will be **particularly significant** in grading this milestone):

- \* after the **proposal** divider:
  - \* your **revised** proposal, if required, clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding your original proposal, and
  - \* your **original proposal**,
  
- \* after the **business rules** divider:
  - \* your latest updated set of business rules, clearly marked with a **Last modified/printed:** date.
  - \* your original set of business rules
  
  - \* (please note --- you are **required** to include a **newly-printed, complete** set of your current business rules in **each** milestone. This does imply that each milestone's preparation should have caused changes in these rules --- even if not, however, you must still include the new set, with an appropriately-changed **Last modified/printed:** date, even if nothing else has changed.)
  
- !!!\* the **third** divider should be neatly labeled, on its tab, as **model**.
- \* after that divider should be your **dated** database model, expressed either as entity-relationship diagrams or as semantic object diagrams. These may be neatly hand-drawn or computer-drawn.
  - \* (however, if they are hand-drawn, you are required keep a **photocopy** of them, in addition to the original turned in --- just in case!)
  
  - \* Remember, for ERD's, that you do not need to include attributes within the diagrams themselves (the ovals are not required...!), but you **do** need to include lists of attributes for each entity separately, either further down the sheet or on the next page (as is called for in our class standards for ERD's).
  
  - \* ERD's need to conform to the given class standard notation (which is slightly different from that given in the course textbook --- be careful!), and semantic object diagrams to the notation given in class and in the course textbook. Indications of **both minimum and maximum cardinality** are required.
  
  - \* Your models need to show if you are meeting the minimum requirements, in terms of number of entities/objects and kinds of relationships, mentioned in the proposal milestone section. In particular, remember that a superclass entity/object and all of its subclass entities/objects count as **one** significant entity/object, all together.
  
  - \* Remember: entities/objects are **NOT** tables or relations. Do **not** refer to them as such, or include information about tables/relations in this section. The project is **not** at the table/relation stage yet!
  
- \* remember: if the previous milestone required that certain changes or additions be made, they must be included here, or this milestone's grade may be affected.

**What to turn in for database design on April 10th (by 1:10 p.m. if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture):**

Your binder, which now contains (!!! is placed before those sections whose contents will be **particularly significant** in grading this milestone):

- \* after the **proposal** divider:
  - \* any **revised** proposal(s), if required, each clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding your original proposal, and
  - \* your **original proposal**,
  
- \* after the **business rules** divider:
  - \* your latest updated set of business rules, clearly marked with a **Last modified/printed:** date. Make sure these include any interrelation constraints.
  - \* your two earlier sets of business rules
  
- \* after the **model** divider:
  - \* a **revised** model, if required, clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding your original model, and
  - \* your **original** database model

Make **sure** that your latest model included here corresponds with the design given in the next section! This does **not** mean that you cram intersection tables, tables due to multi-valued attributes, and other tables resulting from normalization into the model --- tables do **not** belong in your model, only entities/objects! It does mean that if you decide to change the cardinality of a relationship between entities/objects, or if you remove or add an entity/object while doing the design, that you modify the model to reflect that.

!!!\* the **fourth** divider should be neatly labeled, on its tab, as **design**.

- \* after that divider should be your **dated** design, expressed as:
  - \* **nicely formatted SQL create** statements for the tables making up your database
    - \* include neat comments **before** each create statement describing each table's purpose, and explaining any attribute whose meaning is not immediately clear from its name
  
    - \* **primary keys** must be **explicitly** defined for each table
  
    - \* **foreign keys**, as needed, must be **explicitly** defined for all relationships between tables; these should be expressed as **table constraints** rather than column constraints, and should be within the **create table** statement unless there are **unusual** circumstances.
  
    - \* (note that these statements will make clear the **structure** of each table, its **attributes**, and the **"domains"** for each attribute (at least, partial domains...))
  
- \* remember: if the previous milestones required that certain changes or additions be made, they must be included here, or this milestone's grade may be affected.

**What to turn in for database population on April 24th (by 1:10 am if you miss any of that lecture, or by 4:30 p.m. if you attend ALL of that lecture):**

Your binder, which now contains (!!! is placed before those sections whose contents will be **particularly significant** in grading this milestone):

- \* after the **proposal** divider:
  - \* any **revised** proposal(s), if required, each clearly marked with a **Last modified: date**,
  - \* my comment-sheet regarding your original proposal, and
  - \* your **original proposal**,
  
- \* after the **business rules** divider:
  - \* your latest updated set of business rules, clearly marked with a **Last modified/printed: date**. Make sure these include any interrelation constraints.
  - \* your three earlier sets of business rules
  
- \* after the **model** divider:
  - \* any **revised** model(s), if required, each clearly marked with a **Last modified: date**,
  - \* my comment-sheet regarding your original model, and
  - \* your **original** database model

Make **sure** that the latest model given here corresponds with the latest design given in the design section.

- \* after the **design** divider:
  - \* a **revised** design, if required, clearly marked with a **Last modified: date**,
  - \* my comment-sheet regarding your original design,
  - \* your **original** database design

!!!\* the **fifth** divider should be neatly labeled, on its tab, as **contents**.

- \* after that divider should be:
  - \* a **dated** printout of the results of **SELECT \* statements** for **each** of the tables. Each table's select statement(s) should be preceded by a **prompt** command including the **name** of the table. (Note that a **prompt** command is required; a SQL comment will **NOT** suffice.)
    - \* make sure that you do **not** have **echo** set on when you run these!
    - \* use a **monospaced** font for these, please! Select statement results printed using Times-Roman font, for example, are *\*ugly\**. A small font size is preferred over rows that lap several lines, also.
    - \* (note: if you had to change the create table statements given in your design to actually create the tables, you should include the version that actually worked in a **revised design**.)
    - \* (Also, if you have any very-wide tables, please either print them in landscape orientation, or show their contents using several select statements, each including the key and some of the columns.)
    - \* In terms of how much data you should include per table, the main idea is that you should include enough to make your project at least a non-trivial, demonstrable prototype. Trying to give some firmer guidelines, have at least **10** rows per table, in general, with **additional** rows as needed to make your database a reasonable prototype. (Fictitious data is fine.)

- \* remember: if the previous milestones required that certain changes or additions be made, they must be included here, or this milestone's grade may be affected.

### **What to turn in for final version of database project on May 10th (by 4:30 p.m.):**

\* remember: if the previous milestones required that certain changes or additions be made, they must be included here, or this milestone's grade may be affected.

Your binder, which now contains (!!! is placed before those section numbers whose contents will be **particularly significant** in grading this final milestone):

#### **1. Cover Page**

This cover page goes BEFORE the first divider, and must include your name, the course number, the completion date, and some 'title' for your completed database project. You may include additional information as desired.

#### **2. Project Proposal section**

- \* after the **proposal** divider:
  - \* any **revised** proposal(s), if required, each clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding your original proposal, and
  - \* your **original proposal**,

#### **3. Business Rules section**

- \* after the **business rules** divider:
  - \* your final updated set of business rules, clearly marked with a **Last modified/printed:** date. (This final set should include at least 15 business rules for your scenario!)
  - \* your four earlier sets of business rules

Again: yes, as always, I want a newly-printed set, with an updated **Last modified/printed:** date, even if they have not changed since the population milestone.

#### **4. Database Model section**

- \* after the **model** divider:
  - \* any **revised** model(s), if required, each clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding your original model, and
  - \* your **original** database model

Make **sure** that the final model given here corresponds with the final design given in the design section.

#### **5. Database Design section**

- \* after the **design** divider:
  - \* any **revised** design(s), if required, each clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding your original design,
  - \* your **original** database design

#### **6. Database Contents section**

- \* after the **contents** divider:
- \* after that divider should be:
  - \* a printout of the results of **SELECT \*** (or several appropriate SELECT's, as discussed in the population milestone section) from **each** of the final versions of your tables, if they have changed from the previous milestone, clearly marked with a **Last modified:** date,
  - \* my comment-sheet regarding the original population of the tables,
  - \* your "original" population of the tables

\* Note that you may need to add **additional** rows to your tables in order to generate meaningful example queries and reports (see the next two sections).

**(continued on next page)**



### !!!7. Representative Queries section

- \* the **sixth** divider should be neatly labeled, on its tab, as **queries**.
- \* after that divider should be:
  - \* **FIRST a SQL script or scripts, THEN the printed-out results of RUNNING that script or those scripts!!!**

**that is,**

- \* printouts of **script(s)** demonstrating at least **eight** substantially-different and structurally-different representative queries, including at least some "innovative" ones.
  - \* each query should be preceded by a **prompt** command explaining its purpose. This prompt will thus serve as a comment both in the script(s), and in the results of running the script(s). (Note that a **prompt** command is required; a SQL comment will **NOT** suffice.)
  - \* make sure that at least one **join**, at least one **nested query**, at least one appropriate use of an **aggregate function** (such as count(\*), min(), max(), avg(), sum(), etc.) within a select statement, at least one **group-by**, and at least one compound **where** condition (with at least a couple of sub-conditions that are **other than** join-conditions), are included within your set of queries. These must be potentially meaningful/useful to users of your database.
- \* printouts of the **results** of running those queries. Make sure that you include enough sample data so that these results make a meaningful demonstration.
  - \* make sure that you do **not** have **echo** set on when you create these!

These queries should be related to the questions given in your database proposal, although they are not limited to answering those original questions. The **most important criteria** is that they should show **clearly** how the database could be **useful** to end user(s). (That is, your queries should show that your database solves the problem(s) you hoped it would solve, as given in your database proposal.)

### !!!8. Example Reports section

- \* the **seventh** divider should be neatly labeled, on its tab, as **reports**
- \* after that divider should be:
  - \* **FIRST a SQL script or scripts, THEN the printed-out results of RUNNING that script or those scripts!!!**

**that is,**

- \* printouts of **script(s)** demonstrating at least **three** substantially-different and structurally-different reports.
  - \* the code for each report should be preceded by a neat **comment** explaining its purpose. (Yes, this one **is** a SQL comment --- a **prompt** command will **not** suffice for this. You may use **prompt** commands as part of your report design apart from these required comments, however.)
  - \* These should be well-designed and well-laid-out. Human-readability is an important characteristic of a well-designed report.
  - \* "Nice" column formatting --- especially for numeric columns --- is expected, as is "nice" heading formatting. Concatenation should be used to make the reports more pleasant to read --- (for example, instead of having separate first and last name columns, a report should display a single column with the last name concatenated with a comma and then the first name). It is expected that rows will be explicitly ordered in a meaningful way within reports (and this includes appropriate secondary ordering, etc.). At least top titles are expected, also.
  - \* At least one appropriate **break** should be demonstrated, and at least one appropriate **compute** should be demonstrated, amongst your reports. At least two of your reports should be based on queries involving more than one table (or on a view created from more than one table). At least one report should contain at least one column whose contents are appropriate, meaningful numeric data with a well-formatted

**fractional** part.

- \* Breaks should be used to avoid "ugly" repetition in consecutive report rows, and skips should be used **judiciously** to separate results generated using breaks and computes. (Avoid too much skipping of lines also, however; e.g., avoid having a blank line after **every** line in a report.)
- \* printouts of the resulting **reports**. Again, you need to have included enough sample data so that these reports make a meaningful demonstration.
  - \* make sure that you do **not** have echo set on when you create these!

The purpose of these reports is also to demonstrate how your database might be used; they should be designed to be easy to read, and reasonably attractive. They should be related to the hoped-for reports given in your database proposal, if possible, although they are not limited to those original proposed reports.

**!!!9. Discussion section**

- \* the **eighth** divider should be neatly labeled, on its tab, as **discussion**
- \* after that divider should be:
  - \* **Discussion #1: How can this implemented database now be used?**
    - \* You must provide **more than one full page** of discussion about how your particular database, now implemented, can be used.
    - \* Be **specific** about how it may be used to meet some of the goals of your original proposal, or perhaps how it may be used in different ways that you've thought of since writing the proposal.
    - \* This must use **12 pt Times-Roman** font, **double-spaced**, with no more than **1 inch margins** on the top, bottom, left, and right. Center your name on the first line followed by the title on the next line, followed by one blank line, and do not put additional blank lines before or after this. Indent paragraphs, do not put additional blank lines between them. In general, avoid using formatting tricks to "fill" the page --- discussions that do so will be penalized.
  - \* **Discussion #2: How can this implemented database now be maintained?**
    - \* You must provide **more than one full page** of discussion about how your particular database could be maintained over time.
    - \* Do you have relations that must be updated periodically, or based on some event occurrence?
    - \* What kind of issues might arise with regard to keeping the database current and useful over time?
    - \* Would one person be able to take care of this maintenance, or would anyone in a group be permitted to perform maintenance activities, or would a formal database administrator (DBA) be required?
    - \* Again, this must use **12 pt Times-Roman** font, **double-spaced**, with no more than **1 inch margins** on the top, bottom, left, and right. Center your name on the first line followed by the title on the next line, followed by one blank line, and do not put additional blank lines before or after this. Indent paragraphs, do not put additional blank lines between them. In general, avoid using formatting tricks to "fill" the page --- discussions that do so will be penalized.

**10. Source code section**

- \* the **ninth** divider should be neatly labeled, on its tab, as **source code**.
- \* after that divider should be:
  - \* a **pocket** containing, securely, a **diskette**, clearly labeled with your name, containing your implemented database.
  - \* a **printed list** that contains the names of all files on the diskette; after each file name should be a brief description of that file's contents.
  - \* a **separate printed sheet** of instructions for how to set up and use the database (is SQL\*Loader required? which scripts are used to set up and initially populate the database? etc.)

Really, the latter 2 items above would normally be the contents of a README file --- they constitute very simple external documentation for "installing" your project. The diskette must contain all of your SQL scripts and related files (in ASCII form!!) for your completed database, representative queries and example reports, in addition to whatever is needed to create and populate your tables.

(Please note that I may or may not actually run your database --- the notebook should contain printouts of everything you want me to consider while grading your project. However, I need to be able to recreate your database from your diskette, if necessary.)

### 11. Miscellaneous section

- \* the **tenth** divider should be neatly labeled, on its tab, as **misc**.
- \* after that divider should be:
  - \* anything **else** that you wish for me to consider while grading your project --- for example,
    - \* printouts of PL/SQL code for triggers and printouts demonstrating that the triggers work, or
    - \* printouts of forms you have designed and implemented, or
    - \* mention of features you have used in addition to those required (such as sequences, for example, or time/date functions)

If you have made some extra effort, and you want to make sure that I do not overlook it while grading your project, mention it/show it off here!

This final project, if well-done, will be a package you can proudly show off as part of your "portfolio" for interviews, or possibly even use as a reference later for future databases that you design.