

CIS 315 - Project Handout - Fall 2010

When studying the design and implementation of databases, it greatly aids understanding to actually experience the process being studied. To that end, during this semester you will be modeling, designing, and implementing a database for a scenario.

This individual or pair project will use the Oracle RDBMS (Relational Database Management System) on campus. The project consists of a number of milestones, with pieces turned in for each; its final milestone must be turned in by **5:00 pm on Friday, December 10th**. Remember -- like all course work, it will not be accepted late. However, if the final milestone is turned in before **5:00 pm on Thursday, December 9th**, it will be considered **early**, and will receive a **5 point bonus** added to the overall project grade.

Please note -- those working as a pair will be expected to work in a **pair-programming** mode -- that is, you would be expected to work **together** on each milestone, **NOT** splitting up the work and working individually. In pair-programming, **two** people work at the **same** computer, one typing and the other dictating what to type -- neither person is just sitting there, both are thus actively involved in all stages of the project. Do not agree to a pair project unless you are willing to work in this way.

Project 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.**

If you are part of a pair project, note that generally both people involved will receive the same grade. However, I may give peer questionnaires to those working in pairs, and I reserve the right to give different grades to different pair members if I choose, based on my observations, those questionnaires, and other evidence.

Milestone grading breakdowns:

- the overall project grade makes up **20%** of the final course grade, as mentioned in the syllabus.
- the overall project grade is made up of:

Project Milestone	Milestone's Worth	Additional Information
Project Proposal	up to 7.5 points	up to 3 points for the sub-milestone, and up to 4.5 points for the final proposal milestone
Project Model	up to 15 points	up to 3 points for the sub-milestone, and up to 12 points for the final model milestone
Project Design	up to 15 points	
Project Population	up to 7.5 points	
Project Final Milestone	up to 55 points	up to 45 pts baseline, plus up to 10 points for doing more than the minimum (as discussed above)

- (the overall project grade is then the sum of the above, and this overall project grade is multiplied by 0.20 to compute the project portion of your semester course grade.)

Project Proposal Milestone

(worth: up to 3 points + up to 4.5 points = up to 7.5 points)

One important aspect of this project is that you are not just "building a database" -- you are modeling a scenario in which a database could be useful. This distinction is important, and needs to be kept in mind, starting at this Project Proposal phase. Project proposals that are essentially "I want a database to store X", where X is my music collection, or DVD collection, or bird sightings, or watershed measurements, etc., are not suitable for this project, and will not be accepted. You need a scenario with multiple players interacting in various ways -- in fact, you should not be using the words "database", "table", or "relation" at all at this point.

The first milestone has two pieces. First, there is a Project Proposal sub-milestone, in which you submit several paragraphs of scenario description, of the "setting" in which a database might be useful (not of problems or of the desired database, but simply of the "setting" for your eventual database), as well as some introductory questions and business rules (day-to-day rules of operation within that setting). Then, the Project Proposal final milestone will include more-polished description of your proposed scenario, a few more example questions, and a few more business rules.

I will look at your proposal, make sure that it is describing a scenario and not just a desired database, attempt to let you know if it is too broad or too narrow, and then either approve or disapprove it.

Project Proposal sub-milestone (up to 3 points)

Due:

by **5:00 pm** on **Monday, August 30th**

How to submit:

Submit the pieces described below in an e-mail message with the Subject: 315 project scenario.

What to submit:

- Include the name or names of the student(s) making this proposal.
- Write at least **two full paragraphs** that clearly **define and describe** the **scenario** for which you want to design a database. This is not talking about the problems to be solved, nor should it describe the database you want to build. It should describe the **setting** for the eventual database --- this is description, background, backstory --- and it should include description of the people/potential users involved.
- We will be discussing how **business rules** are operational, day-to-day "rules of thumb" in a given setting or scenario. They are user-oriented, NOT database-oriented -- for example, a business might have a business rule that "a 10% down payment is required for special orders", or a club might have a business rule that "the president must attend at least half of all meetings", or a role-playing game might have a business rule that "a player can only carry up to 100 weapons at a time".

Write at least **three** such business rules for your proposed project scenario. Remember to make them day-to-day -operations-oriented, user-oriented, and NOT database-oriented! Write them in the form of a **numbered list**.

- Taking a baby step toward the eventual database - but still **not mentioning databases** yet - it can be useful to consider the kinds of questions users in a scenario would like to be able to ask during their day-to-day operations. For example, in a business setting, one might like to ask, "How many widgets are left in stock?" And in a club, one might like to ask, "In what room is today's meeting?" And for a role-playing game, one might like to be able to ask, "What is my player's current strength level?"

Write at least **two** example questions that users within your scenario might typically ask during a day's work. Write them in the form of a **numbered list** (as a separate list from the list of business rules).

- **PLEASE NOTE:** no movie rental stores or sports-statistics leagues will be accepted as project scenarios. Also please avoid overly graphic or violent project scenarios.

Project Proposal final milestone (up to 4.5 points)

Due:

by **5:00 pm** on **Friday, September 10th**

How to submit:

Submit the files `315proposal.pdf` and `315biz-rules.pdf` using the tool `~st10/315submit` and using a homework number of **11**. (The tool `~st10/315submit` will be discussed in lab before this milestone is due).

What to submit:

REMEMBER: you should **not** specifically be mentioning databases, tables or relations at this point **at all**.

Create a PDF file named `315proposal.pdf` containing the following:

- your name(s)
- CIS 315 - Fall 2010
- the "polished", final version of the scenario and example questions for your proposed database.
 - Remember, the scenario needs to be **at least two full paragraphs** that clearly **define and describe** the **scenario** for which you want to design a prototype database. This is not talking about the problems to be solved, nor should it describe the database you want to build. It should describe the **setting** for the eventual database -- this is description, background, backstory -- and it should include description of the people/users involved.
 - Also include your numbered list of example questions that people within the scenario would like

to be able to ask during their day-to-day operations, during a day's work. You needed at least two of these for the Project Proposal sub-milestone -- you should add to that set so that you have at least **five** of them now.

Create a PDF file named `315biz-rules.pdf` containing the following:

- your name(s)
- CIS 315 - Fall 2010
- the date the business rules were last modified
- your current-as-of-this-point numbered list of business rules for your scenario.
 - remember, these are operational, day-to-day "rules of thumb" for the scenario
 - remember, they are user-oriented, NOT database-oriented
 - you needed at least three of these for the Project Proposal sub-milestone -- you should add to that set so that you have at least **eight** of them now
 - Note that you should be ADDING TO these throughout the project -- you'll submit your dated latest version of your business rules for each project milestone from now on.

Project Model Milestone

(worth: up to 3 points + up to 12 points = up to 15 points)

NOTE: you should **not** be mentioning tables or relations at this point **at all**, nor should you be mentioning primary or foreign keys. Those are appropriate in the next milestone!

Project Model sub-milestone (up to 3 points)

Due:

by **5:00 pm** on **Friday, September 24th**

How to submit:

Submit the pieces described below in an e-mail message with the Subject: 315 model sub-milestone.

What to submit:

E-mail me (in an e-mail with the Subject: **315 model sub-milestone**):

- a list of the entity classes you believe should be part of your database mode at this point.
- an example of a 1:N relationship class you believe should be part of your model -- tell me its name and what entity classes it is between
- an example of a M:N relationship class you believe should be part of your model -- tell me its name and what entity classes it is between
- an example of a multi-valued attribute, and the entity class it is an attribute of, that you believe should be part of your model
- **optionally**, you may also include any questions you have at this point regarding your model-in-progress

Project Model final milestone (up to 12 points)

Due:

by **5:00 pm** on **Monday, October 11th**

How to submit:

Submit the files noted below using `~st10/315submit` and using a homework number of **22**.

What to submit:

Create a PDF file named `315model.pdf` containing the following:

- your name(s)
- CIS 315 - Fall 2010
- the date the model was last modified
- your current-as-of-this-point Entity Relationship Diagram (ERD) for your project, depicting your database model
- make sure this includes the lists of attributes for each entity class, in which multi-valued attributes are indicated, as discussed in lecture
- remember, entity classes are NOT tables or relations. Do NOT refer to them as such, nor include information about tables or relations in the model. The project is NOT at the table/relation stage yet!
 - Likewise, since these are not tables or relations, there are no primary keys or foreign keys yet, either. These are NOT part of the E-R model stage.
- Make sure that your completed model meets the model aspects of the **Project minimum structural requirements** given below.

Also submit the latest version of your business rules, `315biz-rules.pdf`

- Yes, submit it even if it has not changed. (But, hint: you SHOULD find yourself adding to your business rules as part of the modeling process!)
- Be sure that it includes an appropriate last-modified date.

Only submit your `315proposal.pdf` file if it has been **changed** since the Project Proposal final milestone.

- **Note:** if the previous milestone required that certain changes or additions be made to the proposal or to the previous set of business rules, then those changes should be reflected in newly-submitted versions for this milestone, or **this** milestone's grade may be affected

Project minimum structural requirements

Because an important topic in this course is database modeling and design, your project is required to meet certain criteria to help ensure that it is at least somewhat interesting in model and structural senses. **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** entity classes
 - a superclass entity along with all of its subclass entities count as **one** combined entity toward the five-entity-class minimum, unless some of the subtypes have relationships in which only that subtype participates. Be careful, and **ask me** if you have concerns about this.
- at least 4 **distinct, significant** relationship classes, at least one that is **1:N** and at least one that is **M:N**
- at least one **multi-valued** attribute

And, the corresponding **database design/schema** must **correctly implement** all of the above.

Project Design Milestone (worth: up to 15 points)

Due:

by **5:00 pm** on **Friday, November 5th**

How to submit:

Submit the files noted below using `~st10/315submit` and using a homework number of **33**.

What to submit:

Create a SQL script named `315design.sql` containing the following:

- your name(s) (in a comment)
- CIS 315 - Fall 2010 (in a comment)
- the date the design was last modified (in a comment)
- your current-as-of-this-point database schema/design, in which the structure of your database's tables are expressed as **nicely-formatted SQL create table statements**.
 - precede each SQL `create table` statement with a **neat comment** describing the table's **purpose**, explaining any attribute whose meaning is not immediately clear from its name, and elaborating on the domain of any attribute whose logical domain needs more description than is apparent from its physical domain
 - (for convenience later during population, go ahead and also precede each SQL `create table` statement with a corresponding `drop table` statement as well)
 - remember: primary keys must be explicitly specified for each table, and all foreign keys necessary should be explicitly specified as well
 - (thus, these SQL `create table` statements give us 3 of the 4 components of a database design/schema: the tables' structure, their relationships, and at least some indication of each attribute's physical domain, if not its logical domain)
- What about the 4th component of a database design/schema? That's the business rules, and so...

Also submit the latest version of your business rules, `315biz-rules.pdf`:

- Yes, submit it even if it has not changed. (But, hint: you SHOULD find yourself adding to your business rules as part of developing the design/schema!) Also, business rules are officially part of a database design/schema, as you should recall from earlier this semester.
- Be sure that, as always, it includes an appropriate last-modified date.

Also submit the latest version of your model, `315model.pdf`:

- Yes, submit it even if it has not changed. The design/schema should be developed directly from the model, and so one cannot really evaluate the design/schema independent of that model.

- **Make sure** that the model you submit corresponds to your submitted design/schema! Your design/schema will be graded on the basis of the model version that you submit along with it.
 - This does **NOT** mean to add table-related aspects to your model! It means, if you change any maximum or minimum cardinalities, for example, or add or remove any entity classes, etc., based either on comments from me or from insights you get during the design process, that you modify the model to reflect them.
- Be sure that, as always, the submitted model includes an appropriate last-modified date

Only submit your 315proposal.pdf file if it has been changed since the database model milestone.

Additional notes/reminders:

- **PLEASE NOTE:** `insert` statements do **not** belong with the database design. Please save them for the population milestone, and do **not** include them here.
- Remember, if previous milestone(s) required that certain changes or additions be made to the proposal, to the previous set of business rules, and/or to the model, then those changes should be reflected in newly-submitted versions for this milestone, or **this** milestone's grade may be affected

Project Population Milestone (worth: up to 7.5 points)

Due:

by **5:00 pm** on **Friday, November 19th**

How to submit:

Submit the files noted below using `~st10/315submit` and using a homework number of **44**.

What to submit:

Create and submit a `*.sql` file (that is, a SQL script) (and/or `.ctl` and `.dat` files if using SQL*Loader) that populate your tables with some example data (which can be fictitious!)

- include your name(s) in a comment in the SQL script
- include `CIS 315 - Fall 2010` in a comment in the SQL script
- include the date that the SQL script was last modified in a comment within the SQL script
- you may use SQL `insert` statements, or you may use SQL*Loader, your choice!
- how much data should you include? The goal is to 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,
 - ...with additional rows as needed to make your database a reasonable prototype. (You'll find that some tables need additional rows so that others can have sufficient contents.)
 - Again, fictitious data is fine, but make it "look" realistic -- don't use names like 'fg^s&A#', etc.
 - Don't use "real" data that is sensitive! (No "real" social security numbers, credit card numbers, etc.!)

Also create and submit a SQL script `315show-contents.sql` that contains:

- your name(s) (in a comment)
- `CIS 315 - Fall 2010` (in a comment)
- the date the contents script was last modified (in a comment)
- `spool` commands to output its results to a file `315result-contents.txt`
- one or more `select * statements` for each table
 - precede each `select` statement with a `prompt` command to display the name of the table whose rows are being displayed
 - make sure that the rows are readable, and do not "wrap-around" in the `315result-contents.txt` file. There are several ways to achieve this; for example:

- you can show a very-wide table's contents using several `select` statements, each projecting the primary key and a few of its other attributes;
- you can use the `truncate` feature of the SQL*Plus `column` command (ask me if you need help with this)
- you can increase the number of characters printed per line before wrapping by changing `linesize`

Also submit the file `315result-contents.txt` resulting from running `315show-contents.sql`

Also submit the latest version of your business rules, `315biz-rules.pdf`

- Yes, submit it even if it has not changed. (you MAY find yourself adding to your business rules as part of the population process)
- Be sure, as always, that it includes an appropriate last-modified date

Only submit your `315design.sql` file if it has been changed since the database design/schema milestone.

Only submit your `315model.pdf` file if it has been changed since the database design/schema milestone.

- (make sure that your latest model corresponds with your latest database design/schema!)

Only submit your `315proposal.pdf` file if it has been changed since the database design/schema milestone.

Remember: if previous milestone(s) required that certain changes or additions be made to the proposal, to the previous set of business rules, to the model, and/or to the design/schema, then those changes should be reflected in newly-submitted versions for this milestone, or **this** milestone's grade may be affected

Project Final Milestone

(worth: up to 45 points baseline + up to 10 points for work above and beyond = up to 55 points)

Due:

by 5:00 pm on Friday, December 10th

(But remember: the 5 POINT BONUS applies to complete projects submitted by 5:00 pm on Thursday, December 9th)

How to submit:

Submit the files noted below using `~st10/315submit` and using a homework number of 55.

What to submit:

Create and submit a `315queries.sql` file that contains a set of **example queries** using your prototype database, including:

- your name(s) (in a comment)
- CIS 315 - Fall 2010 (in a comment)
- the date the example queries script was last modified (in a comment)
- `spool` commands to output its results to a file `315query-results.txt`
- at least **eight** substantially-different and structurally-different representative queries, including at least some "innovative" ones, that meet the following requirements:
 - each query must be potentially meaningful/useful to users of your database
 - precede each query statement with a `prompt` command explaining the query's purpose and **numbering** the query
 - include at least one **join**
 - include at least one **appropriately-nested** query
 - include at least one appropriate use of an **aggregate function** (such as `count`, `min`, `max`, `avg`, `sum`, etc.)
 - include at least one appropriate use of a `group by` clause
 - include at least one **compound** `where` condition with at least a couple of sub-conditions **other** than join conditions
 - make sure that enough example data is included so that these queries' results are a meaningful demonstration -- you may need to add additional rows for this to be the case.

- These queries could be related to the questions given in your database proposal, but they do not have to be, and you are not limited to answering those original questions.
- The **most important** criterion is that they show **clearly** how the database could be **useful** to end-users within the scenario.

Also submit the file `315query-results.txt` resulting from running `315queries.sql`

Create and submit one or more files `315report1.sql`, `315report2.sql`, ... that create **example reports** from your prototype database, including:

- your name(s) (in a comment)
- CIS 315 - Fall 2010 (in a comment)
- the date that example reports script was last modified (in a comment)
- `spool` commands to output that script's results to a file `315report1-results.txt` (or `315report2-results.txt`, etc.)
- at least three substantially-different and structurally-different representative reports, including at least some "innovative" ones, that meet the following requirements:
 - each report must be potentially meaningful/useful to users of your database
 - the code for each report must be preceded by a neat **comment** explaining its purpose
 - each report should be **well-designed** and **well-laid-out**. Human-readability is an important characteristic of a well-designed report.
 - "nice"/"pretty" column formatting -- especially for numeric columns -- is expected
 - "nice"/"pretty" heading formatting is expected
 - concatenation should be used to make 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 the first name, or vice versa, depending on the report's purpose)
 - rows should be explicitly ordered in a meaningful way within reports (and this includes appropriate secondary ordering as applicable, etc.)
 - at least a top title is expected (for **each** report)
 - include at least one appropriate `break` command whose results are obvious
 - include at least one appropriate `compute` command whose results are obvious
 - 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
 - `break` should be used to avoid "ugly" repetition in consecutive report rows
 - `skip` should be used judiciously to separate results generated using breaks and computes. (Avoid too much skipping of lines, however -- e.g., avoid having a blank line after every row in a

report.)

- make sure that enough example data is included so that these reports' results are a meaningful demonstration -- you may need to add additional rows for this to be the case.
- These reports could be related to the questions given in your database proposal, but they do not have to be, and you are not limited to reports that answer those original questions.
- The **most important** criterion is that they show **clearly** how the database could be **useful** to end-users within the scenario.

Also submit the files `315report1-results.txt`, `315report2-results.txt`, ... resulting from running `315report1.sql`, `315report2.sql`, ...

Create and submit a file `315discussion1.pdf` that contains:

- your name(s)
- CIS 315 - Fall 2010
- the date this first discussion was last modified
- a discussion on "**How can this implemented database now be used?**" that meets the following requirements:
 - it should contain **at least 300 words** (I will measure this using the UNIX `wc` command.)
 - discuss how your particular database, now implemented, can be used within your proposal's scenario
 - be specific!

Create and submit a file `315discussion2.pdf` that contains:

- your name(s)
- CIS 315 - Fall 2010
- the date this second discussion was last modified
- a discussion on "**How can this implemented database now be maintained?**" that meets the following requirements:
 - it should contain **at least 300 words** (I will measure this using the UNIX `wc` command.)
 - discuss how your particular database could be **maintained** over time, again within the context of your proposal's scenario.
 - be specific!
 - do you have relations that must be updated periodically, or based on some event or transaction occurrence?
 - what kinds 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 database? which users would be permitted to

perform maintenance activities? would a formal database administrator (DBA) be required?

Create and submit a file `315readme.pdf` that contains:

- your name(s)
- CIS 315 - Fall 2010
- the date this readme file was last modified
- a list containing names and descriptions of all "code" files (SQL, SQL*Loader, SQL*Plus, etc.) used in the final version of your submitted project
 - that is, follow **each** file name with a brief description of that file's contents
- a **separate** list of instructions for how to set up and use the database (is SQL*Loader required? which scripts are used to create and initially populate the database? etc.)
 - (these two lists are quite common contents in a README file -- they constitute very simple external documentation for "installing" your project)
 - Please note that I may or may not actually run your database -- however, I had better be able to use the instructions above to recreate your database from the files you submit, if necessary.

Also submit the final version of your business rules, `315biz-rules.pdf`

- Yes, submit it even if it has not changed. (you MAY find yourself adding to your business rules as part of the final milestone activities)
- Be sure that, as always, it includes an appropriate last-modified date

Only submit new population-related files if any have been changed since the population milestone.

Only submit your `315design.sql` file if it has been changed since the population milestone.

Only submit your `315model.pdf` file if it has been changed since the population milestone.

- HOWEVER -- NOTE that there will be a SUBSTANTIAL PENALTY if the latest versions of the `315model.pdf` and `315design.sql` files do not correspond/"go together"!!
- ALSO NOTE that there will be SUBSTANTIAL PENALTIES if the **Project minimum structural requirements** given in the Project Model milestone are not met in the latest versions of the `315model.pdf` and `315design.sql` files

Only submit your `315proposal.pdf` file if it has been changed since the population milestone.

Remember: if previous milestone(s) required that certain changes or additions be made to the proposal, to the previous set of business rules, to the model, to the design/schema, and/or to the population, then those changes should be reflected in newly-submitted versions for this milestone, or **this** milestone's grade may be affected

FINALLY, you may, if you wish, also create and submit a file `315misc.pdf`, accompanied by additional files as you desire, that contains:

- your name(s)
- CIS 315 - Fall 2010

- the date this Miscellaneous file was last modified
- a description/discussion/list of anything else that you wish for me to consider while grading your project -- for example,
 - mention of features you have used in addition to those required (such as sequences, or views, or time/date functions, for example) and in which files those are used
 - lists of additional files you are submitting demonstrating additional features or functionality, such as PL/SQL triggers, files demonstrating that the triggers work, code for forms you have designed or implemented, etc.
 - If you have made some **extra** effort, and you want to **make sure** that I do not overlook it while grading your project, **then mention it/show it off here!**

The 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.