**31061/32606 (“5 Assessment”)**

**(Students may be asked to run the code to show that their code is working in the students scheduled lab/lecture time.)**

**Due Check your subject outline.**

**Assessment Item**: The5 Assignment

For the marks for this assessment item to count toward a student’s final grade.

Students **will create a database of their own choosing**,

Students will also write SQL queries for their database.

The 5 assignment is **individual work.** Students may **NOT** work in groups.

Students are required to give a satisfactory presentation about their assignment. The presentations will be marked simply as satisfactory or unsatisfactory.. Prior to giving their respective presentations, each student will hand to the subject coordinator a brief document summarizing their chosen database project.

Late assignment submissions will incur a penalty of 1 mark for each **HOUR** the assignment is overdue. That is, an assignment submitted 1 minute past the due date and time will lose a mark, as would an assignment submitted 59 minutes late. An assignment submitted 60 minutes late will lose 2 marks, and so on. Thus 5 assignments may only be submitted up to one day late. Special consideration, for late submission, must be arranged before the due date with the subject coordinator.

**A very high standard of work is expected from students.**

**Expected numbers of hours to do the assignment**

Students are expected to spend approximately 16-25 hours on this assignment.

**Academic Misconduct**

Students are reminded of the principles laid down in the Faculty’s Statement of Academic Integrity - Good Practice and Ethics in Informal Assessment found at <start.it.uts.edu.au/w/doc/student/integrity.html>, and in particular that:

1. Any collaboration with another person on this assignment should be limited to those described in the “Acceptable Behaviour” section of the Statement of Academic Integrity. Similarly, any group work should be the result of collaboration only within the group.
2. The Faculty penalty for proven and serial misconduct of this nature is zero marks for the Subject. For more information go to; <start.it.uts.edu.au/w/doc/student/integrity.html>.

**Softcopy File Naming**

The two files submitted as attachments should be named as follows (1) family name, (2) other name (both family and other names as shown on your student card), and (3) the name you have given your project. You should give your project a name that is an accurate reflection of the content or purpose of your database. For example***, Joe Blogs 013232322***

**Include your student number** in the file name, or in any part of the softcopy submissions. Your student number should also be provided in the hardcopy submission (where it may be added in legible handwriting).

**Submission Requirements**

The softcopy submission should contain the following:

1. The first page should be the ERD slide from the PowerPoint Presentation – taking up the whole page. The page should also bear your name (as shown on your student card), student number, and the name of the project (these may be handwritten).
2. The complete PowerPoint Presentation as a “handout” (including the ERD slide that also forms page 1 of the hardcopy submission). Six slides to a page is acceptable, provided all writing on the overheads is readable with 6 slides to a page. Otherwise, students should submit 2 slides to a page.
3. A complete listing of the .txt file containing the SQL for the project.

Every page of the softcopy submission should contain your name **AND** **YOUR STUDENT NUMBER** in the document header. **USE YOUR NAME AS SHOWN ON YOUR STUDENT CARD**. We have no way of knowing your other adopted names.

**Marks and Return**

Marks for this assignment will be posted to canvas. Feedback to this assignment is given before the assignment submission.

Students seeking more feedback should contact the subject coordinator after marks have been posted.

**5 Assignment Specification**

Build a database. You decide what database you’d like to build. However, your database must satisfy the “minimal essential” conditions given below, otherwise you will be penalized heavily in the marking.

*Task 0: Choice of Database Project*

Your choice of database should demonstrate some initiative, and be something that is largely your own idea. It must not resemble closely any database you found in the textbook, or any other book, and it must not resemble closely any database built by a student in a previous semester. After you have chosen a project, you should check with the subject coordinator that the project is not too similar to any database in a book or built by a former student. To check with the supervisor:

1. Guidance via Zoom: **communication is preferred, simple approval could be done via zoom, you should show the coordinator your first attempt at the ERD. This may be hand drawn, and may contain errors − it just needs to sufficient to convey to the coordinator what it is you intend to build.**

Unless you have special permission from the subject coordinator (see below), **your database must be inspired by a real web site**. Imaginary data for that web site is acceptable, but real data from that web site is preferred, where possible. You **must** provide the URL for the web site that inspired your project.

Special Permissions: Some students may have a particular application in mind that is not on the world wide web. For example, they may want to build something relevant to where they work. That is acceptable in principle, but students must have the explicit permission of the subject coordinator, **in writing (i.e. email)**. Students are reminded that their assignment submissions are public documents that may be placed in UTSOnline for classmates, and future students to look at. Therefore, the databases for which students seek special permission **should not be populated** with commercially sensitive data.

*The Database Design and the ERD*

**Provide a well designed ERD for this problem, using the notation from the textbook (do NOT provide something in Microsoft Access database diagram format). Indicate the primary key (with an underline) and give the complete attribute list for each entity (i.e. all columns in the given table should occur somewhere in the ERD). If foreign keys are composite, still don’t show them in the ERD unless it is a primary key as well. Include all attributes for each entity in the ERD.**

**Your diagram may be generated with any drawing tool you like, but the ERD must be included in your PowerPoint document (perhaps pasted in as an image). Hand drawn diagrams may be acceptable, provided they are legible (but where the diagrams are illegible, answers will be assumed wrong). Hand drawn diagrams need to be scanned (or digitally photographed) and submitted as part of the PowerPoint document.**

**The database ERD must contain at least: (1) a single one-to-many (or zero to many) relationship, and (2) a single many-to-many relationship, broken into two one-to-many relationships. Failure to meet this criterion will attract zero marks for the entire assignment. However, a database design that just meets this minimal requirement (and meets all other requirements) is likely to score only half the available marks. The complete ERD should fit on a single A4 page, with all writing in the ERD in 12 point.**

***The SQL***

**In PostgreSQL, implement your database design. Populate your database with suitable data for testing the SQL queries you will provide in a subsequent task. Also, provide enough data so that the table rows demonstrate the relationships. For example, if there is a 1:M relationship between 2 tables, ensure that there are at least two records in the M-side table that are related to a respective record in the 1-side table. If the relationship is 0:M, then the appropriate table should contain a row that does correspond to any row in the other table.**

**When submitting the assignment, provide all your SQL, including all the “insert” commands for placing your data into your database.**

**Start the SQL with “DROP” commands for each of your tables, so you can run your script more than once. Use CREATE statements to create your tables. Use “Constraint” to define primary and foreign keys.**

***Domain Integrity:* Where appropriate, add checks on your data.**

***Referential Integrity (1):* Your database should check foreign keys. (2) Your database should also enforce suitable deletion actions.**

When you paste your SQL code into the submission document, use the “Courier New” font. It is a fixed width font, and will preserve your indenting. A variable width font like “Times Roman” will not preserve your indenting.

*Queries*

Write the following queries for your database:

1. A simple query of a single table.

2. A query which uses the words "natural join".

3. The cross product equivalent to the "natural join" query above.

4. A query involving a “Group by”, perhaps also with a “HAVING”.

5. A query which uses a sub query.

6. A cross product which cannot be implemented using the words “natural join” (e.g. self join)

Set out each of these queries as they were set out for the SQL lab exams. That is:

1. An English language description of what the SQL query does.
2. The output generated by it (in Courier New font or another fixed width font).

After all six queries “questions” as specified above, provide the “answers” to all six queries.

**The SQL .txt File: Minimum Essential Conditions**

Your database must satisfy the following “minimal essential” conditions, otherwise you will be penalized heavily in the marking, and perhaps receive zero marks.

1. Your SQL in the .txt file must successfully build a database in PostgreSQL. A SQL submission that produces an error when run within PostgreSQL will attract **zero** marks for the entire assignment. If you can’t get some aspect of the SQL working, leave it out of your project submission.
2. The file containing your SQL should begin with a **comment header block** (i.e. lines beginning with two dashes). The first line of the header block should contain (“5”),Sem, Year”, followed by lines providing your name and email address and your (student number). The header block should then contain, in English, the nature of your database application. Do NOT use technical database language in this section. Write something like the description of the NASA database. You **must** provide the URL for the web site that inspired your project in this heard block.
3. Your SQL should be laid out so it is easy to read. When writing all your SQL, you should approximately follow the indentation style used in the files provided as part of the lab exercises. When you paste your SQL commands into the PowerPoint document, use a fixed-width like Courier to preserve indentation.
4. *Domain Integrity*: You must made good use of “check” statements.
5. *Referential Integrity*: Your SQL should contain all reasonable “ON DELETE” actions, if appropriate both “ON DELETE RESTRICT” and “ON DELETE CASCADE”.
6. All your CREATE statements should precede all your INSERT statements.

**Students are warned that a token effort (i.e. seriously breaches the “minimal essential” conditions) of assignment will attract zero marks.**

**Students are free to produce a database that goes well beyond the requirements of the minimum essential conditions, if they wish to do so for their own satisfaction. However, students are warned that databases that go well beyond the minimum essential conditions do not attract extra marks.**

**PowerPoint Presentation: Minimum Essential Conditions**

The required PowerPoint is a subset of what was required for the Distinction assignment:

1. The first slide should be a title slide, as in the HD assignment/Project. The title page should include the name of the student who produced the slides (and in the **hardcopy only**, his/her student number, which may be hand written).
2. The next slide (or set of slides) should describe the real-world domain modeled by this database. Be succinct. One slide could be enough. The audience just needs to know enough about the domain to understand the ERD.
3. The next slide should show the complete ERD for this database.
4. The next slide should illustrate a single one-to-many relationship in the database (other than a one-to-many relationship that occurs as part of a many-to-many relationship). As in the Distinction PowerPoint presentation, this slide should show both the relationship as shown in the ERD and its realisations in tables, with some example values in the tables.
5. The next slide should illustrate a single many-to-many relationship in the database. As in the Distinction PowerPoint presentation, this slide should show both the relationship as shown in the ERD and its realisations in tables, with some example values in the tables.
6. The next set of slides should show the various queries you are required to write:
   1. A simple query of a single table.
   2. A query which uses the words "natural join".
   3. The cross product equivalent to the "natural join" query above.
   4. A query involving a “Group by”, perhaps also with a “HAVING”.
   5. A query which uses a sub query.
   6. A cross product which cannot be implemented using the words “natural join” (e.g. self join)

For each query, provide (ideally, but not necessarily all on one slide) (1) an English

description of what the query is supposed to find, (2) the actual query, and (3) the

output generated by each of the queries (design queries that are economical in what

they return).

1. The next slide (or set of slides) should illustrate the use of CHECK statements from your SQL. Show a variety of your CHECK statements.
2. The next slide (or set of slides) should illustrate the use of action statements in your SQL. Two examples in the PowerPoint are sufficient. If your database contains both “ON DELETE RESTRICT”and “ON DELETE CASCADE”, give an example of each.
3. The next slide (or set of slides) should illustrate the use of a view in your SQL.