158.337 Group Project Instructions: Part A

You will work in **pairs** for this assignment. Register your group on Assignment Registration link (under Assignment section on the Stream site). Use the following set of steps to guide you in completing the assignment project. You should turn the work in for grading by the due date **(Tuesday, 14th April 4.00 pm).**

Note: Make sure you have altered the password of your allocated Oracle a/c from the given default. You do not want other people accessing your account. Remember it is your responsibility to protect your work.

PART A
Logical and Physical Database Design

Read **Te Ohanga Village - Massey University Student Accommodation Office Case Study** given in the **Appendix A.** For Part A, complete the **three** steps (Step 1 – Step 3).

Part A: Step 1 **(30 marks)**
Create **a logical (relational) data model (e.g. Database or ER diagram).** Use J Developer or Visio to create this diagram.

Use appropriate naming conventions while naming tables, attributes, relationships, etc. in your diagram. Also, use appropriate attribute data-types, data sizes, etc. in the diagram and ensure to flag PKs, FKs, etc.

There should be **two** business rules for each relationship (association) shown on diagram. State these business rules very clearly under a **separate** section (in your report). Similarly, state your **reasonable** assumptions (this helps us better evaluate your design) under a **separate** section.

Examine each table's (entity's) attributes for dependencies and apply the rules of normalisation discussed in Chapter 6 of the textbook. You are required to show the detailed normalisation process {e.g. how you move from lower normal form (e.g. 1NF) to the next (e.g. 2NF)} in the submitted work. Use dependency diagrams for depicting all the relationships among a table's attributes. Normalise tables up to the highest possible NF.

Part A: Step 2 **(20 marks)**
Generate **physical model** based on the logical model created in the Step 1. To create the required set of tables, relationships, constraints, etc. write SQL scripts using a text editor (e.g. Notepad). Use SQL Developer for initial checking / debugging and running of the code. Keep improving your code until you are satisfied with the design.

Check the DDL code in your script file and make sure that all the attribute definitions (e.g. names, data types, data sizes, domain values including defaults, etc.), table constraints (e.g. entity integrity, referential integrity, etc.), etc. are specified correctly.

Validate your work by ensuring that query transactions {(a) - (s) listed towards the end of the case study} are supported by your design. Make sure that the physical design is **robust** i.e. meets the requirements of a good design. Remember normalisation alone does not produce a good design. In the end, ensure that both logical and physical designs are consistent with each other.

Run your final scripts to generate the database (i.e. physical model). Use your Group Oracle a/c to implement the final DDL code.

Part A: Step 3 **(10 marks)**

Finally, create appropriate test data to populate the tables you created in Step2. You can use SQL Developer to run a script file that contains SQL INSERT statements or directly enter data.

There should be reasonably enough rows of the test data across all the tables. However, **no table should contain less than five rows.** Make sure your test data **appropriately and sufficiently reflects** (this would usually mean adding more than 5 rows in some tables) **and does not violate any of the constraints declared** while creating the database. **Set up all the tables and their constraints in the correct order before you load any data into the database.**

Notes:

Apply good naming conventions, which are also self-explanatory (e.g. name create table script file as createTables.sql, name your tables, attributes appropriately, etc.). Additionally, annotate your code (both DDL and DML) with appropriate and enough comments to enhance readability.

Check your project work and make sure that all the scripts run without any errors. Also, ensure that appropriate names have been given to all files.

Assignment Submission

Organise your **project report** (printed and bound) to include all the assignment requirements. Make sure the report contents are also in the order of the laid requirements. There are **three** parts to this assignment submission, a **physical printed report** and two electronic resources - **database and scripts. Put all the files used as solution to Part A into a "single" zip file and submit (one submission per group only) via the Stream Assignment (Part A) link.**

Checklist for physical report submission

1. A logical relational model (**database or ER diagram**) that is laid out **clearly** and **legibly** for grading. The text font size should not be very small (**otherwise risk getting a straight zero) *.** The diagram should show all the entities, attributes (with appropriate data types, sizes, etc.), primary keys, foreign keys and relationship details (appropriate association names, cardinality, strength (identifying-non-identifying), participation (mandatory or optional)), etc. Make sure you have applied and shown normalisation process to arrive at the final model.

2. A **discussion of the reasonable assumptions** you make about the ambiguous aspects and / or reasonable extensions to the case study. We anticipate that **each group's design will differ**** from those of other groups based on the underlying assumptions.

3. A list of **business rules for each relationship** on your diagram (clearly state **two** unambiguous rules for each relationship as these are bi-directional).

4. **DDL Code** i.e. **command file used for creation of your Oracle tables** (remember to include declarative constraints, etc). This shouldn't be auto-generated code and/or linked to your schema account and very much runnable in any Oracle account. Do not use Alter statements to later add the constraints (e.g. referential).

5. **Test Data i.e.** a formatted, readable dump of the **contents of the tables.**

6. **DML Code i.e.** the script file that you use to populate (INSERT commands) the tables. If you directly entered the data into tables, still generate a corresponding sql code file with insert statements and provide in the report.

Checklist for submission under your Group Oracle a/c

Physical design to Part A problem i.e. all the tables with test data must be available via your Group Oracle a/c.

Checklist for Stream submission

All Database scripts for Part A should be uploaded in Stream as a single compressed file under the Assignment link. Include your IDs in the name of your compressed file.

Note:

Enter your names, student id numbers on the project marking sheet **(Appendix B). Submit the completed project report with a marking sheet as a well bound printed document. Make sure to provide the (Group) Oracle username in your report. Submit the script files as a single compressed file to the Stream assignment link. Include your physical database design along with data under your Group Oracle a/c.**

Plagiarism and other project guidelines:

* If print of your diagram were too small then a **ZERO** mark would be awarded. Make sure that the layout of your diagram is good (e.g. avoid many criss-crosses, etc.) and the font size of the entities, attributes etc. is at least Arial 11 (or similar size if using a different font family, style). Make your diagram easy to read for the grader. You might want to print on multiple (A4) pages and glue these together or use bigger sized paper (A3). Please put in effort to make it presentable and readable or risk losing marks.

A **ZERO mark will be given to **ALL** the collaborating parties (no discussion on who did the original work and who copied). Any partial copying will also be awarded a straight ZERO.

Make sure all your work is complete. Graders will **not** be chasing you for the assignment components missing from your submission.

Not adhering to any of the assignment requirements, (e.g. no hard copy) will also be given a straight **ZERO.**

Note: Turnaround time for assignment could be up to three weeks.

Appendix A

Te Ohanga Village, Albany - Massey University Student Accommodation Office Case Study

The Director of *Te Ohanga Village, Albany - Massey University Student Accommodation Office* requires you to design a database to assist with the administration of their accommodation office. The requirements collection and analysis phase of the database design process has provided the following requirements for the *Student Accommodation Office* database followed by examples of the query transactions that must all be supported by the database design.

1. The data stored on each full-time student must include student details such as the identification number, name, home address, date of birth, gender, category of student (e.g. undergraduate, postgraduate), nationality, smoker/non-smoker, special needs, any additional comments, status (placed/waiting), and what degree course the student is studying towards, etc.

   The student information stored relates to those currently renting some form of accommodation and those on the waiting list. Students may rent a room in a hall of residence or a student flat (4-5 bedrooms apartments). Post-graduate, mature may also opt for renting studio units.

   When a student joins the university, he or she is assigned to a staff member who acts as his or her Advisor (of studies). The Advisor is responsible for monitoring the student's welfare and academic progress. The data held for a student's Advisor includes their name, position, name of department, internal telephone number, room number, etc.

2. Each hall of residence has a name (e.g. Tui Hall), address, telephone number, and a hall manager who supervises the operations of the hall. The halls provide only single rooms, which have a room number, place number, and monthly rent rate. Currently, each of the three halls can accommodate up to 70 students. Halls of residence offer fully or partly catered services.

   The place number uniquely identifies each room in all the halls controlled by the student accommodation office and is used when renting a room to a student.

3. The accommodation office also offers student flats. These flats are fully furnished and provide single room accommodation for groups of four to five students. The information held on student flats includes a flat number, address, and the number of single bedrooms available in each flat. The flat number uniquely identifies each flat. Flats are self-catered.

   Each bedroom in a flat has a monthly rent rate, a room number, and a place number. The place number uniquely identifies each room available in all student flats and is used when renting a room to a student.

4. The studio units offered by the accommodation office allow only for single occupancy (i.e. cannot be shared by the students). Mature students with small families usually rent these units (e.g., they can have a partner and/or a child living with them).

   Each studio unit has a monthly rent rate, and a unique studio unit number. In addition, studio units have the fully or partly furnished options but are self-catered.

5. A student may rent a room (in a hall or student flat) or a studio unit for various periods. They need to sign a rent lease agreement before they can move in.

   New lease agreements are negotiated at the start of each academic year with a minimum rental period of one year, which includes semesters 1, 2 and the summer semester. Each individual lease agreement between a student and the accommodation office is required to be unique.

   The data stored on each lease includes the duration of the lease (given as semesters), student details, place number, room number, address details of the hall / student flat / studio, the date the student wishes to enter the room, and the date the student wishes to leave the room (if known), etc.

6. At the start of each semester, each student is sent an invoice for the ensuing rental period. Each invoice has a unique invoice number.

   The data stored on each invoice includes the invoice number, lease number, semester, payment due, student's ID number and full name, place number, room number, and the address of the hall or flat. Additional data is also held on the payment of the invoice and includes the date the invoice was paid, the method of payment (e.g. cheque, cash, etc.), and the date the first or second payment reminder is sent (if necessary), etc.

7. Staff on a regular basis inspect student flats and studio units to ensure that the accommodation is well maintained. The information recorded for each inspection is the name of the member of staff who carried out the inspection, the date of inspection, an indication of whether the property was found to be in a satisfactory condition, any additional comments, etc.

8. Some information is also held on members of staff of the accommodation office and includes the staff number, name, home address, date of birth, sex, position (e.g. Hall Manager, Administrative Assistant, Cleaner, etc.), and location (e.g. accommodation office or Hall).

9. The accommodation office also stores a limited information on the degree courses run by the University such as degree title, degree director name, internal telephone number, and room number, department name, etc. Each student is associated with a single degree course.

10. Whenever possible, information on a student's next-of-kin is stored which includes the name, relationship, contact details, etc.

Query Transactions

The following transactions **must be** supported by your database design. However, you are **NOT required** to implement these transactions (i.e. do not need to write the SQL code for these queries). Use these only to validate your database design.

a) Create a suburb-wise staff list (based on their residential address).
b) Create a report listing students who have their 21st birthday this month.
c) Create a report listing post-graduate students who live in studio units.
d) Create a report listing the Manager's name and telephone number for each of the halls of residence.
e) Create a report listing the names and ID numbers of students with the details of their rent lease agreements.
f) Display the details of lease agreements that include the Semester 1 and 2.
g) Display the details of the total rent paid by a given student.
h) Create a report on students that have not paid their invoices by a given date.
i) Display the details of flat and studio inspections where the property was found to be in an unsatisfactory condition along with the comments.
j) Create a report of the names and ID numbers of students with their room number and place number in a hall of residence.
k) Create a report listing the details of all students currently on the waiting list for accommodation.
l) Display the total number of students in each student category.
m) Create a report listing the names and ID numbers for all students who have not supplied details of their next-of-kin.
n) Create a report listing names and internal telephone numbers of the Advisors along-with their student names.
o) Display the minimum, maximum, and average monthly rent for rooms in halls of residence.
p) Create a report listing all the students that are using part-time catering option in halls of residence.
q) Display the total number of rooms currently occupied and total number of places available in each hall of residence.
r) Display the staff number, name, age, and current location of all members of the accommodation staff who are over 60 years old today.
s) Create staff telephone directory in the order of family name.

Appendix B

158.337 Project Marking Sheet (Part A – 15% course mark)

(Print and attach this page to your project report BEFORE you turn it in.)

(Please make sure you provide all the necessary details)

Oracle Account: Group___

Group Member 1 - ID number, Name

_____

Group Member 2 - ID number, Name

_____

(Grader's section, please do not write below this)

PART A: _____/60marks

Logical Database Design Comments: /30 marks

Physical Database Design Comments: /30 marks

Part A: 15%, Part B: 15% (Total Course Percentage for the project – 30%)