USC

AUSTRALIA

# ICT211 Database Design

# Task 2

**Semester 1, 2019**

# Assessment and Submission Details

## Marks: 30 % of the Total Assessment for the Course

## Due Date: Midnight Friday, 24th May 2019

Assignments are to be submitted by SafeAssign.

**DO NOT SUBMIT THE ASSIGNMENT TO THE COURSE CO-ORDINATOR OR TUTOR.**

Submit your assignment to the link under Assessment -> Task 2 on Blackboard. The submission link will be open a week before the due date. Please follow the submission instructions provided.

The assignment will be marked out of a total of 100 marks and forms 30% of the total assessment for the course. **ALL** assignments will be checked for plagiarism and/or collusion between individuals.

Refer to your Course Outline or the Course Web Site for a copy of the "Student Misconduct, Plagiarism and Collusion" guidelines. Academic Integrity Information.

***Note: Each student MUST be able to produce a copy of their assignment and this copy MUST be produced within 24 hours of it being requested by the Course Co-ordinator. Failure to produce the second copy of the assignment when requested may result in loss of marks or a fail grade for the assignment.***

Requests for an extension to an assignment extension **MUST** be made prior to the date of submission and requests made on the day of submission or after the submission date will only be considered in exceptional circumstances.

# Case Study: Australian Native Plants Ltd – Online CRM Database

## Background

Australian Native Plants Ltd (ANP) is a national plant nursery cooperative specialising in native Australian plants. From its beginnings four years ago in Greenock South Australia, the ANP cooperative now includes six independent plant nurseries around Australia, all specialising in different types of Australian native plants local to their area.

Anna Zikov, the owner of Greenock Australian Native Nursery and founder of ANP, started the cooperative to provide a common national web platform to promote, provide advice and sell Australian native plants. A horticulturalist by profession, Anna enjoys using and experimenting with technology in her nursery and uses sensors and electronic device controllers to automate some of the environment control in her green houses.

Anna enlisted the services of local web developer Josh to build a platform for ANP. Josh had done a great job on the Greenock nursery website and Anna is very pleased with the new ANP website. It is easy to navigate and provides a lot of local information to which other members of the ANP cooperative contribute. She was very surprised that her ANP cooperative idea proved popular with other native plant nurseries and clients. However as more nurseries joined the cooperative, clients and nursery owners started to notice data anomalies such as wrong or out-of-place information, slow retrieval of information, plant order information not reaching the ANP coop member, and random difficulties entering new information.

Anna had a chat with Josh who admitted that, although he is a good web developer, his database design skills were not quite as good. Anna and Josh called on you as a database design consultant to make some recommendations. After some consultation and looking at the database, you realise that Josh had created a flat file database and entity relationships were not logically defined. Your recommendation is to create a Customer Relationship Management (CRM) style database to which nursery industry specific adaptations could be added later. As an experienced database designer, you know that you will not be able to deliver all the database functionality Anna is hoping for in the first project. You explain the need to define a core database system which will provide a foundation for added functionality in the future.

## User Requirements

The following is what you, Josh and Anna agree would be a good core system for this first database design project:

Anna believes the core database entities are coop members, clients, plants and plant orders. You generally agree, but you also know that there will be other regular and composite entities, and business rules which will determine the entity relationships.

Anna explained that coop members will have a unique member id, member name, contact name, date of start, date of end (for when they leave the coop – it will be blank by default), name of nursery, address of nursery, phone, email, and a member description where the ANP member can describe themselves and their specialty.

Josh explained that a client will register on the ANP website and must provide their name, email address, and location. The database must also allocate a client id and start date. The client may order plants so there must be a delivery address attribute as well.

When asked about plants as a database entity Anna explained that each plant entity has a plant id, botanical name, it may have a common name, and a description.

Each plant variety may be stocked by many coop members and each coop member will stock many plants. Each coop member has their own price for each plant variety. This weak entity will have the ANP coop member id and the plant id, price, price date, unit shipping cost (described below for use in a stored function), and an 'in stock' attribute.

Anna went on to explain that clients may order plants from any ANP member. The order will have an order id, client id, ANP member id, order date, order status, shipping date, courier name, shipping cost multiplier, and shipping reference number. Each order must include one or more order items. each order item is a plant with has a plant id, order id, plant cost, quantity, and unit shipping cost.

Because plants can be of various sizes and clients can order plants from any ANP member, shipping costs can vary (for example a client in Queensland may order a rare native plant from Western Australia). To allow the client to get a rough estimate of total costs, including shipping, a stored database function must be created as described below. When a client submits an order through the ANP website, the order is saved into the database and an alert message is sent to the ANP member automatically through the messaging system.

Josh described the messaging system he had built into the ANP website for clients to ask questions and send alerts including notifying ANP members of new client orders, etc. You decide to adjust the messaging system in the database to simplify and make it more efficient as follows: a message entity will have a unique message id, client id, and ANP member id. The message entity will also include a date stamp field and a message field. This simple format will give Josh the flexibility to use it in many different ways on the web site.

## Important requirements

Josh would like to use the sample code that you create. For this reason you MUST:

- Incorporate MySQL database and its corresponding SQL and procedural language,
- Sequence your script so Josh can run (and re-run) it as one sequential script without error

## Required native plant cost calculation stored procedure / function

Clients can purchase native plants from any ANP member. However shipping costs for shipping plants within that ANP member's state is much cheaper than sending them interstate. Anna has a shipping cost agreement among ANP coop members as follows:

- If a client purchases plants from ANP members located within the client's state or territory, the total price for the plant delivered to the client will be the listed price plus the unit shipping cost.
- for all interstate purchases the client will multiply the corresponding shipping multiplier (table below) by the unit shipping cost.

For example:

- Abigail from Parks in New South Wales (NSW) has ordered a quantity of 2 'graceful wattle' from an ANP member nursery in Armadale NSW. The nursery listed price for each 'graceful wattle' is $65 plus $120 shipping cost. The total cost to Harry would be (2 * $65) + (1 * (2 * $120)) = $370 total cost. Shipping Multiplier = 1.
- Johnathan from Perth in Western Australia (WA) has ordered a quantity of 2 'graceful wattle' from the same ANP member nursery in Armadale NSW. The shipping multiplier from NSW to WA is 2.4 so the total cost to Margaret would be (2 * $65) + (2.4 * (2 * $120)) = $706 total cost. Shipping Multiplier = 2.4.

The general calculation is as follows:

*Plant cost = plant quantity * plant price*

*Shipping cost = shipping multiplier * plant quantity * plant unit shipping cost*

*Total order cost = shipping cost + plant cost*

## Required data

**IMPORTANT NOTE** – the following **data MUST be used in your Part B SQL implementation**. A ZERO (0) mark will be awarded for Part B if the following data is not used as part of your SQL implementation.

### *Table 1 - ANP Coop Members*

| Nursery name | Member name | Address |
|---|---|---|
| Native shrubs R us | Joan Coats | 58 Anvidale Street, Armidale NSW 2350 |
| Greenock Australian Native Nursery | Anna Zikov | 37 Greenock Road Greenock 5360 SA |
| Margaret River Natives | Tom Rodicko | 187 Wallcliffe Road Margaret River 6285 WA |
| Tasie Native Plants | Gordon Frost | 423 Lilydale road Launceston 7250 Tas |
| Charnwood Natives | Adriana Codd | 20 Tilliard Drive Charnwood ACT 2615 |
| Territory Native Nursery | Beryl Anthony | 28 Larapinta Drive Alice Springs NT 0870 |

### *Table 2 – Plants*

| Botanical Name | Common Name | Description |
|---|---|---|
| Acacia aneura | mulga | Native tree grows in every state |
| Eucalyptus rhodantha | rose mallee | Native WA tree |
| Goodenia ovata | hop goodenia | Native shrub grows in every state |
| Tetratheca pilosa | pink-eyed susan | Native shrub grows in SA, NT, Tas and Vic |
| Acacia brachybotrya | grey mulga | Native shrub grows in NT, SA and Vic |
| Grevillea arenaria | sand grevillea | Native shrub grows in NT and Qld |
| Verticordia mitchelliana | rapier feather flower | Native shrub grows in WA |
| Banksia repens | creeping banksia | Native shrub grows in WA |
| Eucalyptus saligna | Sydney blue gum | Native tree grows in NSW ans Qld |
| Verticordia plumosa | plumed feather flower | Native shrub grows in WA |
| Tetratheca thymifolia | black-eyed susan | Native shrub grows in Qld, SA and Vic |

## *Table 3 - Shipping Multiplier Table*

| ANP Nursery State / Territory | Destination Client State / Territory | Shipping Cost Multiplier |
|---|---|---|
| New South Wales (NSW) | New South Wales (NSW) | 1 |
| New South Wales (NSW) | Queensland (QLD) | 1.5 |
| New South Wales (NSW) | Northern Territory (NT) | 1.9 |
| New South Wales (NSW) | Victoria (VIC) | 1.2 |
| New South Wales (NSW) | South Australia (SA) | 1.4 |
| New South Wales (NSW) | Australian Capital Territory (ACT) | 1 |
| New South Wales (NSW) | Western Australia (WA) | 2.4 |
| New South Wales (NSW) | Tasmania (TAS) | 2 |
| Queensland (QLD) | Queensland (QLD) | 1 |
| Queensland (QLD) | Northern Territory (NT) | 1.4 |
| Queensland (QLD) | Victoria (VIC) | 1.9 |
| Queensland (QLD) | South Australia (SA) | 1.8 |
| Queensland (QLD) | Australian Capital Territory (ACT) | 1.5 |
| Queensland (QLD) | Western Australia (WA) | 2.5 |
| Queensland (QLD) | Tasmania (TAS) | 2.1 |
| Northern Territory (NT) | Northern Territory (NT) | 1 |
| Northern Territory (NT) | Victoria (VIC) | 2.2 |
| Northern Territory (NT) | South Australia (SA) | 1.2 |
| Northern Territory (NT) | Australian Capital Territory (ACT) | 1.9 |
| Northern Territory (NT) | Western Australia (WA) | 1.5 |
| Northern Territory (NT) | Tasmania (TAS) | 2.5 |
| Victoria (VIC) | Victoria (VIC) | 1 |
| Victoria (VIC) | South Australia (SA) | 1.2 |
| Victoria (VIC) | Australian Capital Territory (ACT) | 1.3 |
| Victoria (VIC) | Western Australia (WA) | 2.2 |
| Victoria (VIC) | Tasmania (TAS) | 1.3 |
| South Australia (SA) | South Australia (SA) | 1 |
| South Australia (SA) | Australian Capital Territory (ACT) | 1.5 |
| South Australia (SA) | Western Australia (WA) | 1.6 |
| South Australia (SA) | Tasmania (TAS) (TAS) | 2.1 |
| Australian Capital Territory (ACT) | Australian Capital Territory (ACT) | 1 |
| Australian Capital Territory (ACT) | Western Australia (WA) | 2.6 |
| Australian Capital Territory (ACT) | Tasmania (TAS) | 1.7 |
| Western Australia (WA) | Western Australia (WA) | 1 |
| Western Australia (WA) | Tasmania (TAS) | 2.4 |
| Tasmania (TAS) | Tasmania (TAS) | 1 |

## Assignment Requirements and Deliverables

**Part A** – Submitted as a MS Word Document:

- Entity Relationship Diagram in Crows Foot Notation
- Relational Schema – including Primary and Foreign Keys
- Supplementary Design requirements – for example but not limited to:
    - information on length of identifiers, postcodes, names,
    - data attribute information (compulsory, variable length / type, etc.)
- Assumptions

**Part B** – Submitted as a single plain text file with name <studentNumber>_crm.sql, containing all your SQL implementation:

**IMPORTANT NOTES:**

- Josh uses a MySQL database. Your Part B MUST work on a MySQL database and be able to be demonstrated to Josh so he is able to apply your SQL implementation into the ANP website.
- Where you are asked to **incorporate the exact data provided** there will be a Zero (0) mark awarded if different data is incorporated.

**Instructions:**

- CREATE TABLE statements for all tables including integrity constraints,
- CREATE TRIGGER statements:
    - Automatically insert a message to the Message table when a client order is placed,
- CREATE FUCNTION / PROCEDURE
    - implement the native plant cost calculation as a function or stored procedure.
- INSERT INTO statements for populating the database:
    - Incorporate the exact 6 nursery ANP member names given in the dataset (make up email addresses and phone numbers)
    - Incorporate the exact 11 plants and their names given in the dataset
    - Incorporate the Shipping Multiplier table into the database as an entity / table - it is recommended to use the abbreviated state name in the database (Vic, Qld, NSW, etc.)
    - Create your own member plant pricing for plants and their shipping costs (at least 3 plants for 3 members)
    - Create at least 3 client entries
    - Create at least 3 client orders
    - Data may need to be inserted in a particular order to comply with integrity constraints,

- SELECT statement/s that will produce the following data for a sample order (you will need to have the data in the database for this query):
    - The Client ORDER will include:
        - client name and account number,
        - order number / id and the total amount for the order,
        - order date,
        - At least three order items:
            - item name,
            - quantity,
            - price,
            - shipping cost,
            - item total cost incorporating the native plant cost calculation Function / Procedure (HINT a function is much easier to call in a SELECT statement)

- SELECT statement that will produce an order report based on a date range and member nursery (you will need to have the data in the database for this query):
    - List of all orders between a start date and an end date for a particular nursery,
    - the report will be grouped by client,
    - each line in the list will include Client name, client state, order number / id and total for that order,
    - will be ordered by oldest order at the top

## Submission

The completed assignment is to be submitted by SafeAssign on or before the due date.

The assignment will be assessed according to the marking sheet (Appendix A). Late submission of the assignment will result in a deduction of 10% of the available marks for each day that the assignment is late (This includes weekends).

## Assignment Return and Release of Grades

Assignment grades will be available on the course web site on two weeks after submission at the latest. An electronic assignment marking sheet will be available.

Where an assignment is undergoing investigation for alleged plagiarism or collusion the grade for the assignment and the assignment will be withheld until the investigation has concluded.

# ICT211 Task 2 – Database Design Report Rubric

| Criteria | High Distinction (85-100%) | Distinction (75-84%) | Credit (65-74%) | Pass (50-64%) | Fail |
|---|---|---|---|---|---|
| **(10%)** 1. Demonstrate an understanding of client requirements. | **Comprehensive and insightful**<br>Client specifications are clearly and comprehensively reflected in the ER diagrams, relational schema, supplementary design requirements. Assumptions show a good depth of insight into the sometimes unclear case study business rules. | **Thorough**<br>Client specifications are comprehensively reflected in the ER diagrams, relational schema, supplementary design requirements. Assumptions show a sound depth of insight into the sometimes unclear case study business rules. | **Effective**<br>Client specifications are clearly reflected in the ER diagrams, relational schema, supplementary design requirements. Assumptions show some insight into the sometimes unclear case study business rules. | **Accurate**<br>Client specifications are generally reflected in the ER diagrams, relational schema, supplementary design requirements and assumptions. | **Narrow / shallow**<br>Client specifications are narrowly / not reflected in the ER diagrams, relational schema, supplementary design requirements and assumptions. |
| **(5%)** 2. Create a cohesive database design that is reflected in the prototype code. | **Comprehensive**<br>ER diagrams, relational schema, supplementary design requirements are comprehensively reflected in the prototype code. Assumptions and business rules are meticulously reflected in the database constraints. | **Thorough**<br>ER diagrams, relational schema, supplementary design requirements are thoroughly reflected in the prototype code. Assumptions and business rules are well reflected in the database constraints. | **Effective**<br>ER diagrams, relational schema, supplementary design requirements are soundly reflected in the prototype code. Assumptions and business rules are mostly | **Accurate**<br>ER diagrams, relational schema, supplementary design requirements and assumptions are generally reflected in the prototype code. | **Narrow / shallow**<br>ER diagrams, relational schema, supplementary design requirements and assumptions are narrowly / not reflected in the prototype code. Some aassumptions and business rules reflected in the database constraints. |

| | | | | | |
|---|---|---|---|---|---|
| | | | reflected in the database constraints. | | |
| **(20%)** 3. Create relational database design schema and documentation. | **Systematic and skillful**<br>Accurate, clear and skillful creation of ER diagrams and relational schema. Systematic, clear and accurate supplementary design requirements and assumptions. | **Thorough and effective**<br>Accurate creation of ER diagrams and relational schema. Thorough and effective supplementary design requirements and assumptions. | **Effective**<br>Effective creation of ER diagrams and relational schema. sound supplementary design requirements and assumptions. | **Competent**<br>Sound creation of ER diagrams and relational schema, supplementary design requirements and assumptions. | **Basic / simplistic**<br>Basic / simplistic creation of ER diagrams and relational schema, supplementary design requirements and assumptions. |
| **(20%)** 4. Create SQL code to create and delete relational database tables. | **Skillful and seamless**<br>The SQL script will be skilfully constructed and seamlessly drop and create MySQL tables along with comprehensive constraints without error. | **Proficient**<br>The SQL script will proficiently drop and create MySQL tables along with sound constraints without error. | **Effective**<br>The SQL script will effectively drop and create MySQL tables along with effective constraints with only minor errors. | **Competent**<br>The basic but sound SQL script will drop and create MySQL tables along with basic constraints with only minor errors. | **Limited / inaccurate**<br>The SQL script has substantial errors / inadequate code when dropping and creating MySQL tables. |
| **(30%)** 5. Create SQL code to insert, search and manipulate the relational database data. | **Skillful and seamless**<br>The SQL script will be skillfully constructed and seamlessly insert, search and manipulate MySQL database data without error. | **Proficient**<br>The SQL script will proficiently insert, search and manipulate MySQL database data without error. | **Effective**<br>The SQL script will effectively insert, search and manipulate MySQL database data with only minor errors. | **Competent**<br>The basic but sound SQL script will insert, search and manipulate MySQL database data with only minor errors. | **Limited / inaccurate**<br>Incorrect data used (Zero mark) The SQL script has substantial errors / inadequate code when inserting, searching and manipulating MySQL database data. |

| | | | | | |
|---|---|---|---|---|---|
| **(15%) 6**. Create SQL code to demonstrate the use and understanding of procedural language in relational databases. | **Skillful and seamless**<br>The SQL script will be skillfully constructed and seamlessly demonstrate MySQL appropriate and correct procedures, functions and / or triggers without error. | **Proficient**<br>The SQL script will proficiently demonstrate MySQL appropriate and correct procedures, functions and / or triggers without error. | **Effective**<br>The SQL script will effectively demonstrate MySQL appropriate and correct procedures, functions and / or triggers with only minor errors. | **Competent**<br>The basic but sound SQL script will demonstrate MySQL appropriate procedures, functions and / or triggers with only minor errors. | **Limited / inaccurate**<br>The SQL script has substantial errors / inadequate code when demonstrating MySQL appropriate procedures, functions and / or triggers. |