



### FIT2094-FIT3171 Databases

2021 Semester 1

### Assignment 2A - SQL - Centre Breeding (CB)

FIT2094 Learning Outcomes: 2, 3, 4, 7 (see Unit Preview)

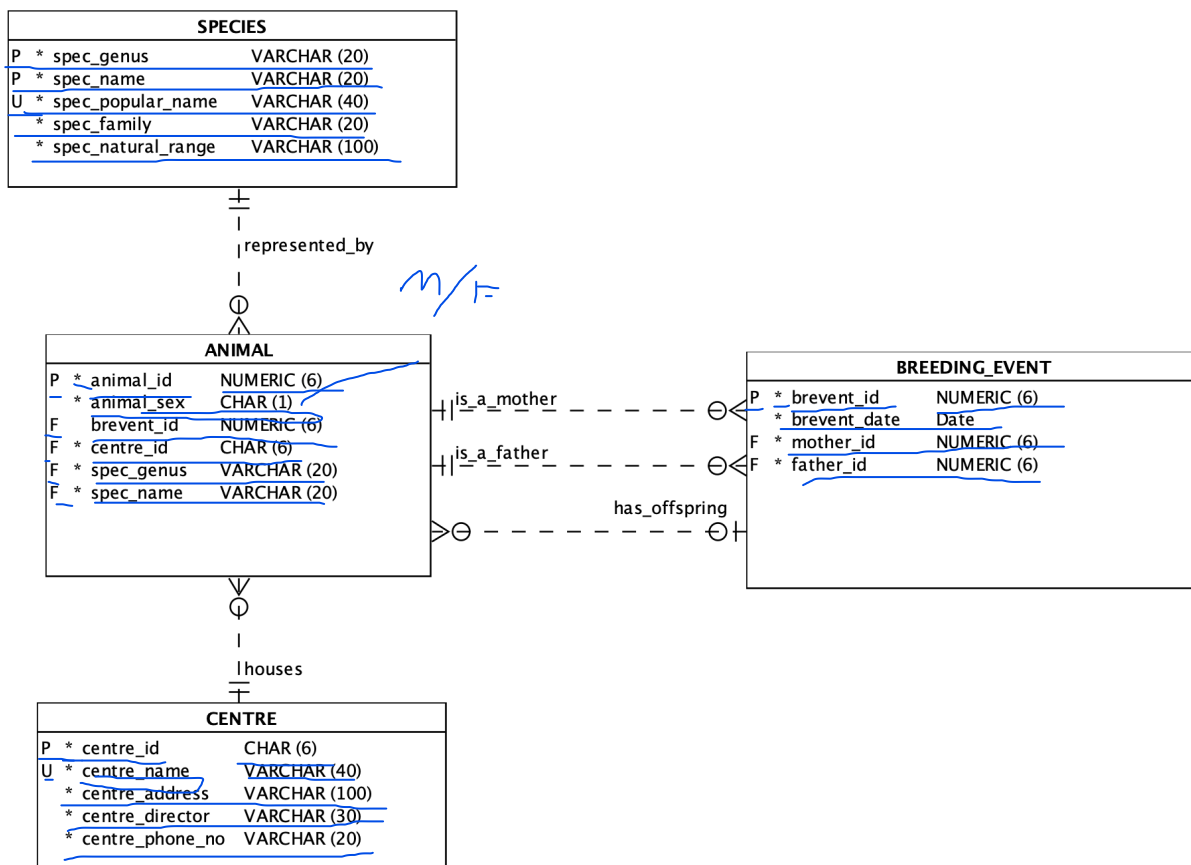
FIT3171 Learning Outcomes: 2, 4, 5, 8 (see Unit Preview)

Assignment weighting 10%

*Assignment marked out of 100 and released as a grade out of 10*

As natural habitats disappear, it is increasingly difficult to ensure that all species can survive in the wild. One solution is to use zoos, reserves and other types of organizations that keep wild animals in captivity as conservation centres for various species. To assist in recording the animals in captivity and their offspring, a Centre Breeding database is to be created in which all centres will record the animals which they hold and the breeding events which take place at the centres.

Database designers have considered the requirements and arrived at the following data model:



## Model Details:

### SPECIES

Records the details of the species, the name of the species, as well as its popular name and the genus and family to which the species belongs. The relevant details for the Tasmanian Devil, for instance, are as follows ...

<b>Genus:</b>	<b><i>Sarcophilus</i></b>
<b>Species:</b>	<b><i>harrisii</i></b>
<b>Popular name:</b>	<b>Tasmanian Devil</b>
<b>Family:</b>	<b>DASYURIDAE</b>

It is also important to know the species' natural range, which is usually given as a description of the geographic regions and natural habitats where a species is found in the wild.

Attribute	Explanation
spec_genus	<u>The species genus</u>
spec_name	<u>The species name</u>
spec_popular_name	<u>The species popular name</u>
spec_family	<u>The species family name</u>
spec_natural range	<u>Description of the natural range of the species</u>

### ANIMAL

Records the details of all animals held at a particular centre

Attribute	Explanation
animal_id	<u>The identifier for the animal</u>
animal_sex	<u>The animal sex (M or F)</u>
brevent_id	<u>If bred in captivity (i.e. at a centre), the id of the breeding event which produced the animal. Animals which have been captured from the wild will have no brevent_id assigned</u>
center_id	<u>The "home" center where the animal is normally located. Animals are sometimes located at other centres for a breeding event</u>
spec_genus	<u>The species genus for the animal</u>
spec_name	<u>The species name for the animal</u>

## CENTRE

Records the details of all centres that keep animals in captivity (zoos, reserves and other types of organizations)

Attribute	Explanation
centre_id	<u>The identifier for the center</u>
centre_name	<u>The centre name</u>
centre_address	<u>The centre address</u>
centre_director	<u>The name of the centres director</u>
centre_phone_no	<u>The centres phone contact number</u>

## BREEDING\_EVENT

Records the details of breeding events which have taken place in the centres

Attribute	Explanation
brevent_id	<u>The identifier for the breeding event</u>
brevent_date	<u>The date on which the breeding event took place</u>
mother_id	<u>The animal_id of the animal who was the mother</u>
father_id	<u>The animal_id of the animal who was the father</u>

## Assignment Tasks

### TASK 1: DDL ( 25 marks):

Using the model and details supplied above, and the supplied **T1-cb-schm.sql** file, create an SQL schema file which can be used to create this database in Oracle.

Note:

- the schema file must be hand coded, not generated in any manner, and
- the schema file must contain:
  - heading documentation including your name and ID
  - drop commands before the create table statements
  - all appropriate constraints, and
  - appropriate column comments

## TASK 2: DML ( 50 marks):

- (a) Using the supplied insert file (**cb-partial-insert.sql**) load the SPECIES and CENTRE tables you have created.
- (b) Load the ANIMAL and BREEDING\_EVENT tables with **your own test data** using the supplied **T2-cb-insert.sql** file script file, and SQL commands which will insert *as a minimum*, the following sample data -
- 12 animals, some of which must have been captured from the wild, i.e. are not the offspring from a breeding event, and
  - 4 breeding events.

In adding this data you must ensure that the test data thoroughly tests the model as supplied, so as to ensure your schema is correct.

Your inserted data must conform to the following rules:

- You may treat all of the data that you add as a single transaction since you are setting up the initial test state for the database.
- The primary key values for this data should be hardcoded values (i.e. **NOT** make use of sequences) and must consist of values below 100.
- Dates used must be chosen between the 1st January 2020 and 31st January 2021.
- Animals with the species popular name of 'Tasmanian Devil' must not be added to the centres AUS10 or AUS20

For this task **ONLY**, Task 2(b), you may look up and include values for the loaded tables/data directly where required. However, if you wish, you can still use SQL to get any non-key values.

**In carrying out this task you must not modify any data or add any further data to the tables which were populated by the cb-partial-insert.sql script.**

**[ 25 marks]**

For all subsequent questions (Task 2(c) onwards) **you are NOT permitted to:**

- manually lookup a value in the database, obtain its primary key or the highest/lowest value in a column,
- manually calculate values external to the database, e.g. on a calculator and then use such values in your answers. Any necessary calculations must be carried out as part of your SQL code, or
- assume any particular contents in the database - rows in a table are potentially in a constant state of change

**You must ONLY use the data as provided in the text of the questions. Failure to adhere to this requirement will result in a mark of 0 for the relevant question.**

For the following tasks, **your SQL must correctly manage transactions and use sequences to generate new primary keys for numeric primary key values** (under **no** circumstances may a **new primary** key value be **hardcoded** as a number or value).

(c) Your answers for these tasks (task 2c) must be placed in the supplied SQL Script **T2-cb-dm.sql**

- (i) Create sequences which will allow entry of data into the ANIMAL and BREEDING EVENT tables - the sequences must begin at 500 and go up in steps of 1 (i.e. the first value is 500, the next is 501 and so on)

**[ 2 marks]**

- (ii) It has been decided to close the 'SanWild Wildlife Sanctuary' and transfer all the animals to the 'Kruger National Park', make these changes to the database including removing the closed centre.

**[5 marks]**

- (iii) The 'Australia Zoo' acquired its first 'Tasmanian Devil' animal, a female (F), which was acquired on the 2nd February 2021. On the same date, 'Werribee Open Range Zoo' also acquired its first 'Tasmanian Devil' animal, a male (M). Both of these animals were acquired from the wild (not bred). Add these two animals to the database. 'Tasmanian Devil' is the animal's species popular name.

**[ 6 marks]**

- (iv) These two animals have been used as a breeding pair at the 'Australia Zoo', the breeding event took place on the '10 Feb 2021'. Immediately after the breeding event, the male animal was returned to the 'Werribee Open Range Zoo' (this movement does not need to be recorded - see the centre\_id details in the ANIMAL table above). Twenty eight days later two young were born to the mother at the 'Australia Zoo', one a female (F) and the other a male (M). Add these details to the database. In adding these details it should be noted that both the 'Australia Zoo' and the 'Werribee Open Range Zoo' each still only have a single 'Tasmanian Devil', the one they acquired on the 2nd February 2021 and that neither animal has been involved in any other breeding event to date.

**[12 marks]**

### TASK 3: DATABASE MODIFICATIONS ( 25 marks):

Your answers for these tasks (task 3) must be placed in the supplied SQL Script

#### T3-cb-alter.sql

The required changes must be made to the "live" database (the database after you have completed tasks 1 and 2) **not** by editing and executing your schema file again. Before carrying out the work below, please ensure that you have completed tasks 1 and 2 above.

- (a) The various **centres** have decided that they would like to be able to record the **total number of offspring which are born at each centre**. Code the SQL to add an attribute to the database which would allow this to be recorded as part of the system. Initially, and also be **default all counts should be set to 0**. You are *not* required to populate this attribute simply add it to the database, it will be populated by a subsequent SQL command which you are not required to supply.

**[3 marks]**

- (b) **The animals in this system will eventually die (they are all alive at the moment). What is an appropriate approach to deal with this situation? Update the database to implement your chosen approach. Select any animal in your data which was bred in a centre (you may manually select a PK value here) and provide the SQL commands to demonstrate how your approach works. In a comment block briefly explain your approach and the advantages of this approach.**

**[8 marks]**

- (c) **The type of each center is to be recorded as part of the database. Modify the database so that for each centre a type can be recorded. The standard types to be assigned are:**

- Zoo
- Wildlife Park
- Sanctuary
- Nature Reserve
- Other

The **default type** for a center should be set as **'Other'**. You should note, in arriving at your solution, that it may be necessary to **extend/modify these types at regular intervals**.

Update the database to assign the relevant type to each current centre based on the following criteria using the centres name:

- if the name contains 'Zoo' assign the type Zoo
- if the name contains 'Park' assign the type Wildlife Park
- if the name contains 'Sanctuary' assign the type Sanctuary
- if the name contains 'Reserve' assign the type Nature Reserve

**[14 marks]**

## SUBMISSION REQUIREMENTS

**Due Date: Wednesday 12th May 2021 at 5 PM AEST (week 10)**

*Please note, if you need to resubmit, you **cannot** depend on your tutors' availability, for this reason, please be **VERY CAREFUL** with your submission. It is strongly recommended that you submit several hours before this time to avoid such issues.*

For this assignment there are four files you are **required** to submit:

- T1-cb-schm.sql
- T2-cb-insert.sql
- T2-cb-dm.sql
- T3-cb-alter.sql

If you need to make any comments to your marker/tutor please place them at the head of each of your solution scripts in the "Comments for your marker:" section.

**Do not zip these files into one zip archive**, submit four independent SQL scripts. The individual files must also have been pushed to the FIT GitLab server with an appropriate history as you developed your solutions (a minimum of four pushes - 1 per file, however we would *strongly recommend more than this*). Please ensure your commit comments are meaningful.


***Late submission will incur penalties at the rate of -5 mark for every 12 hours the submission is late.***

Please note we **cannot mark any work on the GitLab Server**, you need to ensure that you submit correctly via Moodle since it is only in this process that you complete the required student declaration without which work **cannot be assessed**.

**It is your responsibility to ENSURE that the files you submit are the correct files - we strongly recommend after uploading a submission, and prior to actually submitting, that you download the submission and double-check its contents.**

Your assignment **MUST** show a status of "Submitted for grading" before it will be marked.

## Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading 
Grading status	Not graded

If your submission shows a status of "Draft (not submitted)" it will not be assessed and **will incur late penalties after the due date/time**.

Please **carefully** read the documentation under the "Assignment Submission" on the Moodle Assessments page which covers things such as extensions and resubmission.

## CRITERIA FOR MARKING

Submissions will be graded on:

- the correct application of relational database principles,
- the correct handling of transactions and the setting of appropriate transaction boundaries i.e. correct placement of commits, and
- the correct application of SQL statements and constructs to:
  - populate tables,
  - modify existing data in tables, and
  - modify the "live" database structure to meet the expressed requirements (including appropriate use of constraints). In making these modifications there must be no loss of existing data or data integrity within the database.

Submissions **will be grade penalised** if they:

- contain SET ECHO ... or SPOOL commands
- make use of views
- use subqueries and SQL conditions unnecessarily,
- do not use to\_char/to\_date where appropriate in handling dates,
- do not have an appropriate development history on the FIT GitLab server for all source files (**at least four pushes required**).