FIT2094 Databases

Assignment 1 - Database Design

Ocean Odyssey

Purpose	Given the provided case study, students will be asked to transform the information provided into a sound database design and implement it in Oracle. This task covers learning outcomes: 1. **Apply the theories of the relational database model.** 2. **Develop a sound relational database design.** 3. **Implement a relational database based on a sound database design.**
Your task	**This is an open-book, individual task. The output for this task will be an initial conceptual model as a PDF document and a logical model implemented in the Oracle RDBMS**
Value	**40** % of your total marks for the unit
Due Date	**Wed, 30 April 2025 at 4:30 pm**
Submission	● Via Moodle Assignment Submission. ● **FIT GitLab check-ins will be used to assess the history of development**
Assessment Criteria	● Using the supplied case study description prepare a conceptual model identifying the required entities, attributes and relationships. ● Normalise the supplied case study forms/s and integrate the resultant relations into a logical model derived from the identified conceptual model. ● Depict the data requirements expressed in the case study via a relational database logical model. ● Generate a schema that meets the case study data requirements from the logical model produced ● Consistent use of industry-standard notation and convention
Late Penalties	● 5% of the marks available for the task (-5 marks) deduction per calendar day or part thereof for up to one week ● Submissions over 7 calendar days after the due date will receive a mark of zero (0), and no assessment feedback will be provided.
Support Resources	See Moodle Assessment page
Feedback	Feedback will be provided on student work via: ● general cohort performance ● specific student feedback 15 working days post-submission (approved by ADE) ● a sample solution

Case Scenario

Ocean Odyssey (OO) is a worldwide travel company. The company books passengers on ships that operate cruises departing from various ports worldwide. Each ship is operated by a particular company known as the operator. Each operator is assigned an operator ID as an identifier, and the company's name and the Chief Executive Officer's name are recorded. A given operator operates one or more ships. For each ship, Ocean Odyssey records a ship code to identify the ship, the ship's name, the date the ship was commissioned, its tonnage, its maximum guest capacity and the name of the country where the ship is registered.

The cabins on a given ship are identified by a cabin number (such numbers may be reused across ships, e.g. many ships may have a cabin D1). Ocean Odyssey records a particular cabin's sleeping capacity and the cabin's class for a given ship (this class classifies the quality of the experience and services available).

A cruise uses a particular ship (a cruise only uses one ship) and departs on a particular date and at a particular time. A cruise ID identifies each such cruise. Ocean Odyssey records the name of the cruise and a brief description of the cruise.

Passengers register with Ocean Odyssey when they make their first cruise booking. Each passenger is assigned a unique ID. The passenger's first and last name is recorded. Ocean Odyssey also records the passenger's gender and date of birth. If the passenger is a minor (i.e. under 18 years of age), another registered passenger must be designated as a guardian. The guardian must be able to be identified by the system. This data is used during booking to ensure minors are accompanied by their guardian.

Each passenger's address is recorded as a street (including street number), town, postcode, and country. When the members of a particular family book on a cruise, they often all have the same address.

Ocean Odyssey maintains a manifest (list of booked passengers) for all cruises they manage. This manifest records the cabin that has been allocated for each passenger for each cruise (this allocation is carried out when the passenger is booked on the cruise). For each passenger taking part in a cruise, OO also records the date and time when they first boarded the ship.

REMEMBER to keep up to date with the Moodle Ed Assignment 1 forum, where further clarifications may be posted (this forum is to be treated as your client).

To view Assignment 1 only posts, select the Assignment and then the Assignment 1 forum from the Categories list in the left panel.
Once selected, you can Filter the posts via the Filter option at the top of the list of posts:
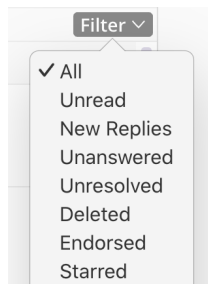
Please be careful to ensure you do not publicly post anything that includes your reasoning, logic, or any part of your work to this forum. Doing so violates Monash plagiarism/ collusion rules and carries significant academic penalties. Use private posts to raise questions that may reveal part of your reasoning or solution.

You are free to make assumptions if needed; however, they must align with the details here and in the assignment forums and must be clearly documented (see the required submission files). **Normally, such assumptions would only relate to minimum cardinality, which was not expressed in the case study**.

GIT STORAGE

Your work for these tasks **MUST** be saved in the *provided* Assignment/Ass1 folder of your local repository and *regularly pushed to the FIT GitLab server* to build a clear history of the development of your model.

TASKS to be Completed

TASK 1 Ocean Odyssey Conceptual Model [15 Marks]

Based on the case scenario on page 2 of this document, prepare a **CONCEPTUAL** model for Ocean Odyssey. In preparing this model, *you must only use the description provided on page 2 of this document*. Your model must be saved in a file named **oo_conceptual.pdf**

Your development history, as pushed to GitLab, must clearly show the steps you have been taught:

- Step 1: entities and keys
- Step 2: relationships, and
- Step 3: non-key attributes

The PDF file of your model must have at least three pushes (remember all pushes must be of a file with the same name - oo_conceptual.pdf). Please note that three pushes are a minimum; you are free to make more (and we would expect more, in which case you will have more than one commit/push for each step). You must regularly check that your pushes have been successful by logging in to the FIT GitLab server's web interface; you must not simply assume they are working. **Do not forget to check that your GitLab author details are correct for every push**. Before submission via Moodle, you must log in to the GitLab server's web interface and ensure your final submission files are present.

GIT automatically maintains a history of all files pushed to the server. You do not need to, and MUST not, add a version name to your various versions. Please ensure you use the same name (oo_conceptual.pdf) for all saved versions of your solution.

The steps to complete this task:

Using LucidChart, prepare a **FULL conceptual model** (Entity Relationship Diagram) using crow's foot notation for Ocean Odyssey (OO) as described above.

- For this FULL conceptual model (ERD), include:
  - Identifiers (**keys**) for each entity
  - all required attributes and
  - all relationships. Cardinality (min and max) and connectivity for all relationships must be shown on the diagram.

- **Surrogate keys must not be added to this model**.

Your model must conform to the unit ERD standards listed in the "Conceptual Modelling" Applied lesson "Unit Entity Relationship Diagram Standards" on Ed. Your name must be shown on your diagram, and it must be exported as an A4 portrait page.

TASK 2 Ocean Odyssey Normalisation [15 marks]

The image below shows *two* sample cruise itineraries:

Cruise ID   1234
Cruise Name   Sydney Brisbane Getaway
Cost per person $810

Port Code	Port Name	Port Latitude	Port Longtitude	Country code	Country Name	Departure/Arrival	Date/Time
AUSYD	Sydney	-33.865143	151.209900	AUS	Australia	Depart	19 Feb 2025 15:30
AUBNE	Brisbane	-27.470125	153.021072	AUS	Australia	Arrive	21 Feb 2025 07:00
AUBNE	Brisbane	-27.470125	153.021072	AUS	Australia	Depart	21 Feb 2025 17:00
AUSYD	Sydney	-33.865143	151.209900	AUS	Australia	Arrive	23 Feb 2025 08:00

Cruise ID   1432
Cruise Name   Northern Adventure
Cost per person $2,490

Port Code	Port Name	Port Latitude	Port Longtitude	Country code	Country Name	Departure/Arrival	Date/Time
ISREY	Reykjavik	64.128288	-21.827774	ISL	Iceland	Depart	10 Mar 2025 16:00
ISGRF	Grundarfjordur	48.780239	-3.04608	ISL	Iceland	Arrive	11 Mar 2025 07:00
ISGRF	Grundarfjordur	48.780239	-3.04608	ISL	Iceland	Depart	11 Mar 2025 19:00
ISAKU	Akureyri	65.688492	-18.126169	ISL	Iceland	Arrive	12 Mar 2025 08:00
ISAKU	Akureyri	65.688492	-18.126169	ISL	Iceland	Depart	12 Mar 2025 19:00
GLJFR	Paamiut	61.828800	-160.215851	GRL	Greenland	Arrive	15 Mar 2025 07:00
GLJFR	Paamiut	61.828800	-160.215851	GRL	Greenland	Depart	15 Mar 2025 16:00
ISREY	Reykjavik	64.128288	-21.827774	ISL	Iceland	Arrive	19 Mar 2025 06:00

Note that a cruise may "loop" around its origination port, i.e. depart from the origination port, return to the originating report and then depart again, all as part of the same cruise.

Perform **normalisation to 3NF** for the data depicted in the supplied sample documents (*note there are two samples here; you only need to normalise one document/itinerary*). This normalisation must be based only on the depicted form content - you must not introduce attributes not shown on the document.

The approach **you must use** is shown in the "Normalisation" Applied class solutions. You must begin by representing the document you are working on as a single UNF relation and then move through 1NF, 2NF, and 3NF. **No marks will be awarded if you use a different approach**.

During normalisation, you must:

○   **Do not add surrogate keys**.
○   **Include all attributes shown on the form** (you must **not remove** any attribute as derivable)
○   Clearly **show UNF, 1NF, 2NF and 3NF**.
○   Clearly **show all candidate keys** for each relation in 1NF.
○   **Identify the Primary Key in all relations by underlining the PK attribute/s**.
○   **Identify all dependencies** at the various normalisation stages (Partial at 1NF, Transitive at 2NF and Full at 3NF). You should use the same notation as depicted in the normalisation sample solutions, for example:

> attr1 -> attr2, attr3

   If none exist, you must note this by stating:

> *No partial dependencies present* and/or
> *No transitive dependencies present*

○   **Carry out attribute synthesis if required**.

The **relation and attribute names used throughout your normalisation and those on your subsequent logical model must be the same**.

Your normalisation must be completed in an MS Word, Apple Pages, or Google document with a filename of **oo_normalisation**.

If using MS Word or Pages, place the source document inside your local Assignment 1 GitLab repo (Assignments/Ass1). The source document must be regularly saved and pushed to GitLab as you develop your normalisation.

If you are using a Google document, you must regularly download the normalisation as a file called oo_normalisation.pdf and push it to GitLab. You must maintain the source Google document and make it available to your marker on request.

Your normalisation must have at least three pushes (remember all pushes must be of a file with the same name - oo_normalisation) to GitLab. The file extension for oo_normalisation will depend on which software you choose to use. Ensure that your name is shown on every page of the normalisation.

TASK 3 Ocean Odyssey Logical Model [55 marks]

Ocean Odyssey has supplied some further information to guide your modelling:
- The company records each passenger's contact phone number; for minors, no contact number will be recorded (the contact for their guardian will be used). The phone number should be recorded as a simple attribute. A new entity should not be created to hold the phone number.
- Cabins across the various ships are assigned a cabin class as one of the following:
  - Interior
  - Ocean view
  - Balcony, or
  - Suite
  These classes are fixed and will not be modified.

1. Prepare a logical level design for the Ocean Odyssey database based on your Task 1 Conceptual model, the normalisations you carried out in Task 2 above and further details supplied here in Task 3.

   - The logical model must be drawn using the Oracle Data Modeler. Information engineering or Crow's foot notation must be used to draw the model. Your logical model must **not** show data types. **You must create a new empty folder in your local repo, in the Ass1 folder, called oo_model, and then place your model inside this folder, naming the saved model as oo_logical.**
   - All relations depicted must be in 3NF. Candidate keys are possible natural keys; **you must ensure your model protects all candidate keys** to maintain the business rules.
   - You must **add at least one surrogate key to your design** (you are free to select the most appropriate relation to make this change in). You **must explain why you added the surrogate key to your chosen relation as part of your assumptions**. We have a unit rule about requiring a surrogate key if the relation has a composite key with more than two attributes, but this is not the only reason you might add a surrogate. You may add surrogate keys to multiple relations if you wish.
   - All **attributes must be commented** *in the database* (i.e., the comments must be part of the table structure, not simply comments in the schema file).
   - Check clauses/look-up tables must be applied to attributes where appropriate.
   - You **MUST** include the legend in your model. Please edit the legend panel to show your name and ID number.
   - **Please carefully check the slide "Overall Design Process - checklist" from the "Logical Modelling" Workshop and ensure you follow the steps listed.**
   - Your GIT repository must indicate your development history with multiple commits/pushes as you work on your model. **A minimum of six pushes is required for your logical model** *as it is developed* **to show this history. You are free to make more pushes/commits and are encouraged to do so.**

2. **Generate the database schema in Oracle Data Modeler and use the schema to create the database in your Oracle account**.

The *only* edit you are permitted to carry out to the generated schema file is to add header comment/s containing your details and the commands to spool/echo your run of the script (as illustrated in "Logical Modelling" Applied Stage 3 on Ed). In generating your schema file, ensure you:

- Capture the output of the run of your schema statements using the spool command.
- Ensure your script includes drop table statements at the start of the script.
- Name the schema file as **oo_schema.sql.**

Please note when working with your model, **ensure that you NEVER select any export options from the Data Modeler menu**:
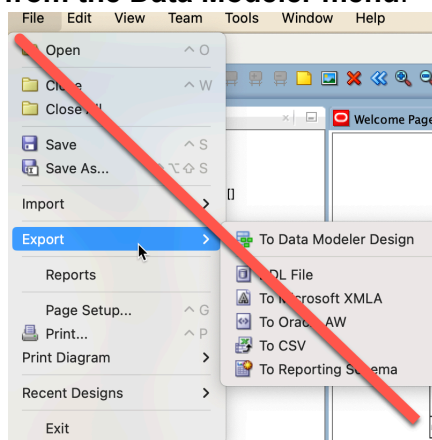
such actions can fill your Oracle account space and render it unusable.

Tasks 1, 2 and 3 - Use of Modeling Standards/Meeting Submission Requirements and Git usage [15 marks]

See the Marking Guide section of this document for further details.

Use of Generative AI tools

In this assessment, you can only use generative artificial intelligence (AI) to assist with design decisions. **Any use of generative AI must be appropriately acknowledged** (see Learn HQ)

Requirements

The following seven files are to be submitted and **must also exist** in your FITGitLab server repo:

- A **single-page pdf file** containing your full final conceptual model. Name the file **oo_conceptual.pdf**. This file must be created via File - Export (or Download As) - PDF from LucidChart (**do not use screen capture**) and must be able to be accessed with a development history via GIT. You can create this development history by downloading your PDFs (**don't forget to use the same name, oo_conceptual.pdf - DO NOT use version 1, etc**) and committing/pushing to GIT as you work on your model. In exporting from LucidChart, **please** **select a page size of A4 with portrait mode**.

- A **PDF** document showing your full normalisation of the sample cruise itineraries showing all normal forms (UNF, 1NF, 2NF and 3NF). Name the file **oo_normalisation.pdf**

- A single-page **PDF** file containing the final logical Model you created in Oracle Data Modeler. Name the file **oo_logical.pdf.** This pdf must be created via File - Data Modeler - Print Diagram - To PDF File from within Data Modeler, do not use screen capture.

- A zip file containing your Oracle Data Modeler project (when zipping these files, be sure to include the .dmd file and the folder of the same name). Name the zip file **oo_model.zip.**

  Part of the assessment of your submission will involve your marker extracting your model from this zip, opening it in Data Modeller, and engineering to a new Relational model. From this, your marker will generate a schema, which will then be compared with your submitted schema (**they must be the same for your schema to be accepted**). For this reason, your **model must be able to be opened by your marker and contain your <u>complete model</u> (i.e. both your logical <u>and</u> relational models);** *otherwise, your submission will not be able to be fully marked, resulting in a significant loss of marks*. You **MUST** carefully check that your model is complete - ensure you take your submission archive, **copy it to a new temporary folder**, extract your submission parts, extract your model and ensure it opens correctly *before* submission. Please view the video on Ed under the lesson "A6 Oracle Data Modeler Support Videos", which demonstrates this process.

- A schema file (CREATE TABLE statements) generated by Oracle Data Modeler. Name the file **oo_schema.sql**

- The output from the Oracle spool command showing the tables have been created. Name the file **oo_schema_output.txt**

- A PDF document containing any assumptions you wish to make your marker aware of. Name the file **oo_assumptions.pdf**. If you have made no assumptions, submit the document with a single statement saying, "No assumptions made".

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***.

Academic Integrity

Students are expected to be familiar with the University Academic Integrity Policy and are particularly reminded of the following:
Section 1.9:
Students are responsible for their own good academic practice and must:
• undertake their studies and research responsibly and with honesty and integrity;
• credit the work of others and seek permission to use that work where required;
• not plagiarise, cheat or falsify their work;
• ensure that their work is not falsified;
• not resubmit any assessment they have previously submitted, without the permission of the chief examiner; appropriately acknowledge the work of others;
• take reasonable steps to ensure that other students are unable to copy or misuse their work; and
• be aware of and comply with University regulations, policies and procedures relating to academic integrity.
and
Section 2.9:
Unauthorised distribution of course-related materials: Students are not permitted to share, sell or pass on to another person or entity external to Monash:

2.9.1 any course material produced by Monash University (such as lecture slides, lecture recordings, class handouts, assessment requirements, examination questions; excluding Handbook entries) as this is a breach of the Copyright Compliance Policy and such conduct may be a copyright law infringement subject to legal action; or

2.9.2 any course-related material produced by students themselves or other students (such as class notes, past assignments), nor to receive such material, without the permission of the chief examiner.

The penalties for breaches of academic misconduct, include:

- a zero mark for the assessment task
- a zero mark for the unit
- suspension from the course
- exclusion from the University.

Where a penalty or disciplinary action is applied, the outcome is recorded and kept for seven years, or for 15 years if the penalty was exclusion.

Marking Guide

Submitted models will be assessed against the optimal solution for this modelling task.
After Assignment 1 has been graded, this optimal solution will be available as a sample solution.
Your submission will be graded out of a possible 100 marks,

Marking Criteria	Items assessed
TASK 1 [15 marks] Conceptual Model	
Identification of the entities that support the case study **[5 marks]**	**Maximum 5 marks - Entities and Keys:** ● Marks awarded for each correct entity identified ● Mark penalty for unnecessary entities included ● Marks awarded for each correct key selected ● Mark penalty for surrogate or foreign keys added
Identification and placement of attributes to support the case study **[5 marks]**	**Maximum 5 marks - Attributes:** ● Marks awarded for each necessary attribute identified ● Mark penalty for extra attributes included ● Marks penalty for placement of attribute in incorrect entity
Determination of relationships that support the case study **[5 marks]**	**Maximum 5 marks - Relationships and Cardinality:** ● Marks awarded for each correct relationship identified ● Mark penalty for unnecessary relationships included ● Mark penalty for redundant relationships included ● Marks awarded for correct minimum and maximum cardinality for every correct relationship
TASK 2 [15 marks] Normalisation	
Normalise the supplied case study documents and integrate the resultant relations into a logical model. **[15 marks]**	**Maximum 12 marks - Normalisation:** ● Marks awarded for each correct normalisation step ● Marks awarded for correct attribute synthesis ● Mark penalty for additional attributes or surrogate keys added during normalisation **Maximum 3 marks - Dependency diagrams:** ● Marks awarded for each correct dependency depicted within normalisation **Mapping to logical model:** ● Mark penalty for incorrect mapping of each relation to logical model
TASK 3 [55 marks] Logical Model	
Depict the data requirements expressed in the case study via a relational database logical model. **[45 marks]**	**Maximum 10 marks - Relations:** ● Marks awarded for each required relation and its attributes identified ● Mark penalty for extra relations included

- Marks penalty for placement of attribute in incorrect relation
- Mark penalty for multivalued attributes included

Maximum 10 marks - Primary keys:

- Marks awarded for each correct assignment of a primary key

Maximum 10 marks - Relationships:

- Marks awarded for each required relationship identified
- Mark penalty for each incorrect minimum and maximum cardinality for each required relationship depicted
- Mark penalty for unnecessary relationships included
- Mark penalty for redundant relationships included

Maximum 4 marks - Surrogate key:

- Marks awarded for creation of at least one appropriate surrogate key
- Marks awarded for creation of unique constraint/s to protect natural key/s

Maximum 5 marks - Attribute data types:

- Marks awarded for each correctly identified Oracle data type
- Marks awarded for each null constraint correctly implemented based on business rules

Maximum 6 marks - Business Rules:

- Marks awarded for each correctly identified integrity requirement to implement case study's business rules

Generate a schema which meets the case study data requirements from the logical model produced **[10 marks]** **This section will be awarded 0 marks if no relational model is present in your submission - please CHECK your model carefully.**	**Maximum 5 marks - Relational model generation: this section** - Marks awarded for correct generation of relational model from submitted logical model. - *MUST be presented in submitted model and must match generation from the submitted logical model* **Maximum 5 marks - Schema generation:** - Marks awarded for correct generation of SQL schema file from submitted logical model - Mark penalty applied for any errors generated when script run in Oracle - Mark penalty for missing column comments
GENERAL [15 marks]	

Consistent use of industry standard notation and convention Met submission requirements **[5 marks]**	**Maximum 5 marks - Modelling standards and submission requirements:** ● Marks awarded for application of Unit Conceptual and logical model notation conventions ● Mark penalty for showing data types on logical model ● Mark penalty for missing model legend and/or relationship labels ● Mark penalty for missing required documents and/or not adding student name to all pages submitted ● Mark penalty for submitting zipped file

Correct use of Git
[10 marks]
	Maximum 10 marks - Git used appropriately:
	
	● Marks awarded for a minimum of twelve pushes showing a clear development history as follows
	   ○ Conceptual model 3 pushes
	   ○ Normalisation 3 pushes, and
	   ○ Logical model 6 pushes
	● Marks awarded for correct Git author details used in pushes (see Unit Software Installation - Git Pull, Add, Commit and Push in your Ed lesson)
	● Marks awarded for the use of *meaningful commit messages* (ie, not blank or of the form "Push1")

Late submission penalty -5 marks for each 24 hours late or part thereof