FIT2094 Databases

Creating, Populating and Manipulating Database - World Cruise (WC)

Purpose	Students will be asked to implement, via SQL, a small database in the Oracle RDBMS from a provided logical model case study, followed by the insert of appropriate data to the created tables. Once populated the database will be used to: carry out specified DML commands and make specified changes to the database structure via SQL. This task covers learning outcomes: 1. Apply the theories of the relational database model. 3. Implement a relational database based on a sound database design. 4. Manage data that meets user requirements, including queries and transactions.
Your task	This is an open book, **individual task**. The final output for this task will be a set of tables and data implemented in the Oracle RDBMS
Value	**25%** of your total marks for the unit
Due Date	Wednesday, **12th October 2022**, 4:30 PM (AEDT) / 1:30 PM (MYT)
Submission	● SQL Portfolio (weekly submission week 8 - week 11) via Moodle Assignment Submission ● Via Moodle Assignment Submission ● FIT GitLab check ins will be used to assess history of development
Assessment Criteria	● Application of relational database principles. ● Handling of transactions and the setting of appropriate transaction boundaries. ● Application of SQL statements and constructs to create and alter tables including the required constraints and column comments, populate tables, modify existing data in tables, and modify the "live" database structure to meet the expressed requirements (including appropriate use of constraints).
Late Penalties	● 10% deduction per calendar day or part thereof for up to one week ● Submissions more than 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 ten working days post submission ● a sample solution

INSTRUCTIONS

World Cruises books passengers on ships which host cruises. Each ship is operated by a particular company known as the operator. Each operator is assigned an operator id as an identifier and has the company's name and Chief Executive Officer's name recorded. A given operator may operate one or more ships. For each ship, World Cruises records a ship code to identify the ship, the ship's name, the date the ship was commissioned, the ship's tonnage, its maximum guest capacity and the name of the country in which the ship is registered.

The cabins on a given ship are identified by a cabin number (such numbers may be reused across ships eg. many ships may have a cabin number 211). World Cruises records for a given ship, the capacity of a particular cabin and the class of the cabin (this class classifies the quality of the experience and services available). The cabin class is one of the following: interior, ocean view, balcony, or suite.

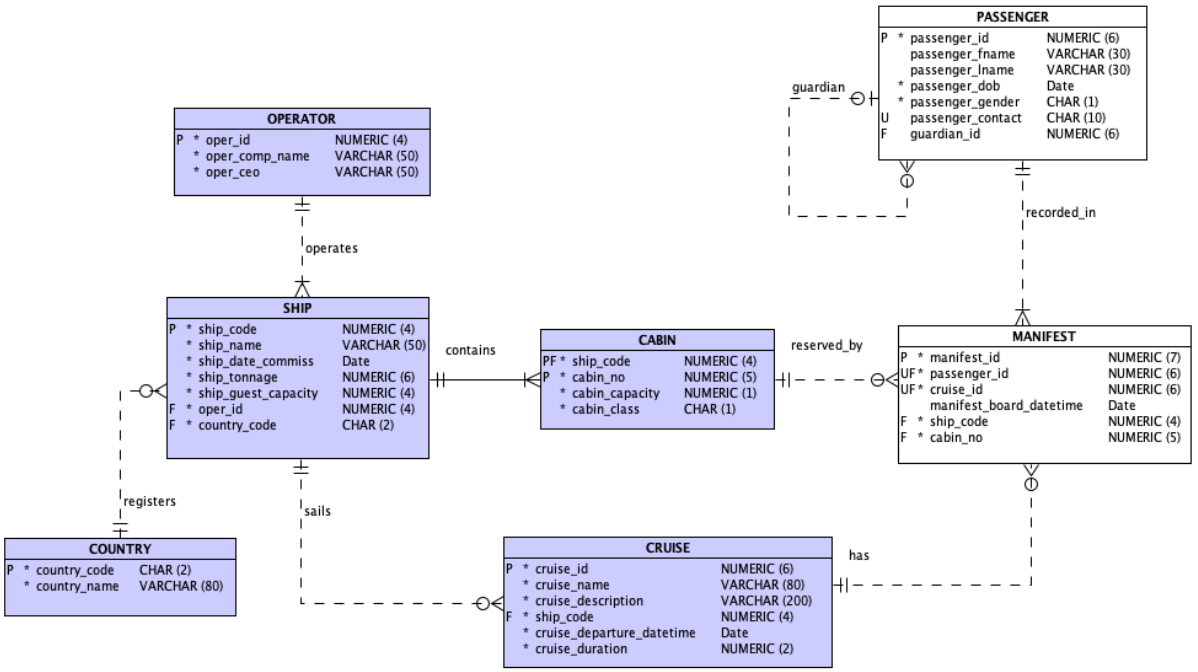
A cruise makes use of a particular ship (a cruise only uses one ship) and departs on a particular date and time. Each such cruise is identified by a cruise id. World Cruises records the name of the cruise, a brief description, and the duration (in days) of the cruise.

Passengers register with World Cruises when they make their first cruise booking. Each passenger is assigned a unique id. The passenger's name is recorded as first name and last name. World Cruise also records the passenger's gender and date of birth. If the passenger is a minor (ie. under 18 years of age), a booking can only be accepted if another passenger on the same cruise can act as a guardian. The guardian must be identified by the system.

World Cruises record for each passenger their contact phone number, for a minor no contact number will be recorded, the contact for their guardian will be used.

World Cruises maintains a manifest for all cruises they manage. This manifest records for each cruise, the cabin which has been allocated for each passenger (this allocation is carried out when the passenger is booked on the cruise). All cruises board passengers only at the cruise's originating port on the scheduled departure date and time. For each passenger taking part in a cruise, WC also records the date and time when they first boarded the ship. Passengers may board up to 24 hours earlier than the scheduled departure.

A model to represent this system has been developed:



The schema/insert file for creating this model (wc_schema_insert.sql) is available in the archive ass2_student.zip - this file partially creates the World Cruise tables and populates several of the tables (those shown in purple on the supplied model) - you should read this schema carefully and be sure you understand the various data requirements. **You must not alter the schema file in any manner**, it must be used as supplied.

Steps for working on Assignment 2

1. Download the Assignment 2 Required Files (ass2_student.zip) archive from Moodle
2. Extract the zip archive and place the contained files in your local repository in the folder /Assignments/Ass2. Do not add the zip archive to your local repo. Then add, commit and push them to the FITGitLab server.
3. Run wc_schema_insert.sql
4. Write your answer for each task in its respective file (eg. write your answer for Task 1 in T1-wc-schema.sql and so on).
5. Save, add, commit and push the file/s regularly while you are working on the assignment
6. Finally, when you have completed all tasks, upload all required files from your local repository to Moodle. Check that the files you have uploaded are the correct files (download them from Moodle into a temporary folder and check they are correct). After you are sure they are correct, submit your assignment. Note that the **filenames must not be changed** - you must submit files with the same names as those supplied in the supplied archive.

The final SQL scripts you submit MUST NOT contain SPOOL or ECHO commands (you may include them as you work but must comment them out before submission). Please carefully read the Marking Guide on pages 11 and 12.

TASKS

ENSURE your **id and name are shown at the top of any file you submit**.

GIT STORAGE
Your work for these tasks MUST be saved in your individual local working directory (repo) in the Assignment 2 folder and regularly pushed to the FIT GitLab server to build a clear history of development of your approach. Any submission with less than eight pushes to the FITGitLab server will incur a grade penalty of 10 marks. Please note eight pushes is a *minimum*, in practice we would expect significantly more.

Before submission via Moodle, you **must** log into the web interface of the GitLab server and ensure your files are present in your individual repo and that their names are unchanged.

TASK 1: DDL (16 marks)

For this task you are required to add to **T1-wc-schema.sql**, the CREATE TABLE and CONSTRAINT definitions which are missing from the supplied partial schema script in the positions indicated by the comments in the script.

The table below provides details of the meaning of the attributes in the missing two tables. You **MUST use identical relation and attribute names** as shown in the data model above to name the tables and attributes which you add. The attributes **must be in the same order** as shown in the model. You must use delete RESTRICT/NO ACTION for all FK constraints. These new DDL commands *must be hand-coded, not generated in any manner (generated code will not be marked)*.

Table name	Attribute name	Meaning
PASSENGER		
	passenger_id	Unique identifier for a passenger
	passenger_fname	Passenger first name
	passenger_lname	Passenger last name
	passenger_dob	Passenger date of birth
	passenger_gender	Passenger gender (M for male, F for female, or X for non-binary/indeterminate/intersex/unspecified/other)
	passenger_contact	Passenger contact number
MANIFEST		
	manifest_id	Unique identifier for a manifest
	manifest_board_datetime	Date/time passenger boarded ship

To test your code you will need to first run the provided script **wc_schema_insert.sql** to create the other required tables. **wc_schema_insert.sql**, at the head of the file, contains the drop commands for ***all*** tables in this model. If you have problems with Task 1 and/or Task 2 simply rerun **wc_schema_insert.sql** which will cause all tables to be dropped and correct the issues in your script. **Do not add DROP TABLE statements** to either of your Task 1 or Task 2 scripts.

TASK 2: Populate Sample Data (24 marks)

Before proceeding with Task 2, you must ensure you have run the file wc_schema_insert.sql (which **must not be edited in any way**) followed by the extra definitions that you added in Task 1 above (T1-wc-schema.sql).

Load the PASSENGER and MANIFEST tables with **your own test data** using the supplied **T2-wc-insert.sql** script file, and SQL commands which will insert ***as a minimum***, the following sample data:

(i) 15 PASSENGER entries
- Included at least 5 passengers who are under 18 years of age

(ii) 30 MANIFEST entries
- Included at least 10 passengers
  - out of these 10 passengers, at least 4 of them are under 18 years of age
- Included at least 5 cruises which uses at last 3 different ships
- Have at least 2 passengers who completed more than 1 cruise
- Have at least 2 passengers who did not show up
- Have at least 2 passengers who book future cruises

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

Your inserted data must conform to the following rules:

(i) You may treat all the data that you add as a single transaction since you are setting up the initial test state for the database.

(ii) The primary key values for this data should be hardcoded values (ie. **NOT** make use of sequences) and must consist of values below 100.

(iii) The data added must be sensible (eg. boarding date time should be within 24 hours before the cruise scheduled departure).

For this task **ONLY,** Task 2, 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 *wc_schema_insert.sql* script.

For all subsequent questions (Task 3 onwards) **you are NOT permitted to**:

- manually lookup an attribute/s in the database to obtain *any* value,
- manually calculate values (including dates/times) 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 contents in the database - rows in a table are potentially in a constant state of change

Your answers must recognise the fact that you have been given, with the supplied insert file, only a small sample snapshot of a multiuser database, as such you must operate on the basis that there will be ***more data in all the tables of the database than you have been given. Your answers must work regardless of the extra quantity of this extra "real" data and the fact that multiple users will be operating in the tables at the same time. You must take this aspect into consideration when writing SQL statements.***

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**.

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).

TASK 3: DML (24 marks)

Your answers for this task (Task 3) must be placed in the supplied SQL Script **T3-wc-dm.sql**

For this task you are required to complete the following sub-tasks in the same order they are listed. Where you have been supplied with a string contained in quotes, such as 'New Zealand Delight' you may search in the database using the string *as listed*. Where a particular case (upper case, lower case, etc.) for a word is provided you must only use that case. When a name is supplied you may break the name into first name and last name, for example 'Dominik Davies' can be split into 'Dominik' and 'Davies', again note that the case must be maintained as it was supplied.

(a) Oracle sequences are going to be implemented in the database for the subsequent insertion of records into the database for the PASSENGER and MANIFEST tables.

Provide the CREATE SEQUENCE statement to create two sequences which could be used to provide primary key values for the PASSENGER and MANIFEST tables. Both sequences should start at 100 and increment by 1. Immediately prior to the create sequence commands, place appropriate DROP SEQUENCE commands so they will cause the sequences to be dropped before being created if they exist. *Please note that there can only be these two sequences introduced and used in Task 3.*

[2 marks]

Question 3b, 3c and 3d are related questions. You can use the information given in 3b to answer 3c and 3d

(b) A family, consisting of a father named ‘Dominik Davies’ and his two daughters (‘Henrietta Davies’ and ’Poppy Davies’) booked a ‘Melbourne to Auckland’ cruise which will depart on 23 October 2022 at 10:00AM.

This is the first time the family has booked a cruise run by World Cruise. They booked a balcony cabin number 210. Dominik’s contact number is ‘0493336312’. Both children are under 18 years old and do not have their own contact numbers. Their father is their guardian.

Make these changes to the data in the database. You may assume that there is only one cruise named ‘Melbourne to Auckland’ departing on 23 October 2022 at 10:00AM in the system and you may make up sensible data for the rest of attributes. This entire cruise booking should be treated as a single transaction.

[8 marks]

(c) A week later, Poppy was unwell and required a complete rest. Her father called World Cruise to change their booking for the ‘Melbourne to Auckland’ departing on 23 October 2022 at 10:00AM cruise.

Dominik informed World Cruise that he wanted to cancel the booking for ‘Poppy Davies’. He also wanted to downsize the cabin to an ocean view cabin with 2 people capacity. He and Henrietta were then assigned cabin number 113.

Make these changes to the data in the database. You may assume that Dominik Davies is only listed as a guardian for his two kids in the system.

[8 marks]

(d) A few days after the booking alteration, Dominik Davies decided to cancel the booking altogether.

Make these changes to the data in the database.

[6 marks]

TASK 4: DATABASE MODIFICATIONS (20 marks)

Your answers for these tasks (Task 4) must be placed in the supplied SQL script **T4-wc-alter.sql**

The required changes must be made to the "live" database (the database *after* you have completed tasks 1, 2 and 3) **not** by editing and executing your schema file again. Before carrying out the work below, please ensure that you have completed tasks 1, 2 and 3 above.

If in answering these questions you need to create a table, please place a drop table statement prior to your create table statement.

(a) WC wants to track the number of passengers booked on each cruise. Add a new attribute in the CRUISE table to record this and initialise the attribute with the correct number of passengers based on the data which is currently stored in the system. If there is no passenger booking for a particular cruise, then the value of this new attribute must be set to 0.

Modify the database structure to meet this new requirement.

[4 marks]

(b) The ships used for cruises require regular maintenance. WC wish to store the ship maintenance records from this point forward. For each ship, WC wants to store the maintenance start date, maintenance expected end date, and maintenance type. There are three maintenance types used by WC: Preventive or Scheduled Maintenance, Corrective or Breakdown Maintenance, and Condition Maintenance. These types will not be expanded in the near future.

Modify the database structure to meet this new requirement.

[8 marks]

(c) WC have realised that their database model has a serious weakness; a minor may travel with different guardians on different cruises. In the current model, if the guardian is changed (guardian_id in the passenger table), the result will be incorrect historical data - the previous cruise records may not refer to the correct passenger who acted as a guardian during the cruise. To address this weakness, WC wishes to record the guardian of a minor for *each cruise* that the minor is booked on (i.e. record the guardian history).

Modify the database structure to meet this new requirement. Note that you must not lose current guardian details for previous cruises which are stored in the database. You may assume that the nominated guardian in the passenger table was on board for all currently recorded cruises.

[8 marks]

Submission Requirements

Due Date: **Wednesday, 12th October 2022 at 4:30 PM AEDT / 1:30 PM MYT**

*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-wc-schema.sql
- T2-wc-insert.sql
- T3-wc-dm.sql
- T4-wc-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 eight pushes - 2 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 -10 marks for every 24 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 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.

Resubmission

If you wish to resubmit your assignment you must email your tutor, provide your full details as listed below and request that they reopen your submission for a second submission. Note if this resubmission is after the due date/time the submission will be regarded as late.

When you contact your tutor (or workshop leader) via email, please ensure you clearly include your full name, unit code and applied class number as part of every email you send so they can identify who the message has come from. This will ensure we can respond as quickly and accurately as possible.

You must NOT assume that your tutor will be available if you require a resubmission close to the due date/time - they may have classes or not be available for other reasons, so do not leave submission to the very last minute.

Marking Guide

Submitted code will be assessed against an optimal solution for this task - this optimal solution will be available as a sample solution after Assignment 2 has been graded. Given that this is SQL there are often several alternative approaches possible, such alternatives will be graded based on the code successfully meeting the briefs requirements. If it does, the answer will be accepted and graded appropriately.

Marking Criteria	**Items Assessed**
TASK 1 DDL 16 marks	
DDL Creation of tables	**Maximum 8 marks - Create table:** ● Marks awarded for correct table DDL ● Marks awarded for correct attributes/data types ● Marks awarded for correct PK definition ● Marks awarded for correct use of column comments ● Mark penalty applied if generated schema used
DDL implementation of non-PK database constraints	**Maximum 8 marks - Non-PK Constraints:** ● Marks awarded for correct implementation of non-PK constraints
TASK 2 Populate Sample Data 24 marks	
Insert of required items test data	**Maximum 12 marks - Insert of data:** ● Marks awarded for correct insert of required data ○ 15 PASSENGER entries ○ 30 MANIFEST entries ● Marks awarded for correct management of transactions
Insert of valid test data	**Maximum 12 marks - Valid data inserted:** ● Marks awarded for validity of data inserted ○ meets the requirements expressed in the assignment brief ● Marks awarded for correct management of dates when inserting

Task 3 DML 24 marks	
	Maximum 24 marks - Satisfy brief requirements: ● Marks awarded (a) - (d) for SQL code which meets the expressed requirement ● Mark penalty applied if commit not used appropriately ● Mark penalty applied if date handling and string database lookups not managed correctly
Task 4 Database Modifications 20 marks	
	Maximum 20 marks - Satisfy brief requirements: ● Marks awarded (a) - (c) for SQL code which meets the expressed requirement (including appropriate use of constraints). In making these modifications there must be no loss of existing data or data integrity within the database. ● Mark penalty applied if commit not used appropriately ● Mark penalty applied if column comments not used where required
Penalty Criteria	**Penalty Applied**
Limited/No push of model to FITGitLab server resulting in lack of development history.	If less than eight pushes showing a clear development history a ***grade deduction of 8 marks applied***. Note that the expectation is that you would push significantly more times than this.
Use of ● VIEWs ● SET ECHO or SPOOL commands, and/or ● PL/SQL	Use of VIEWS, inclusion of SET ECHO/SPOOL and/or PL/SQL commands in submitted scripts will result in a ***grade deduction of 8 marks being applied.***
Incorrect application of relational database principles	Marks will be deducted, based on any question, where basic relational model principles have been violated. For example, creation of a table which is not in 3NF