



MONASH University

Information Technology

FIT3171 Introduction to Databases

2021 Semester 1

Assignment 2B - SQL - MonHeli (MH)

Version 1.1 20th May 2021

Changes marked with highlight

Learning Outcomes: 2, 5, 6, 8 (see Unit Preview)

Assignment weighting 15%

Assignment marked out of 100 and released as a grade out of 15

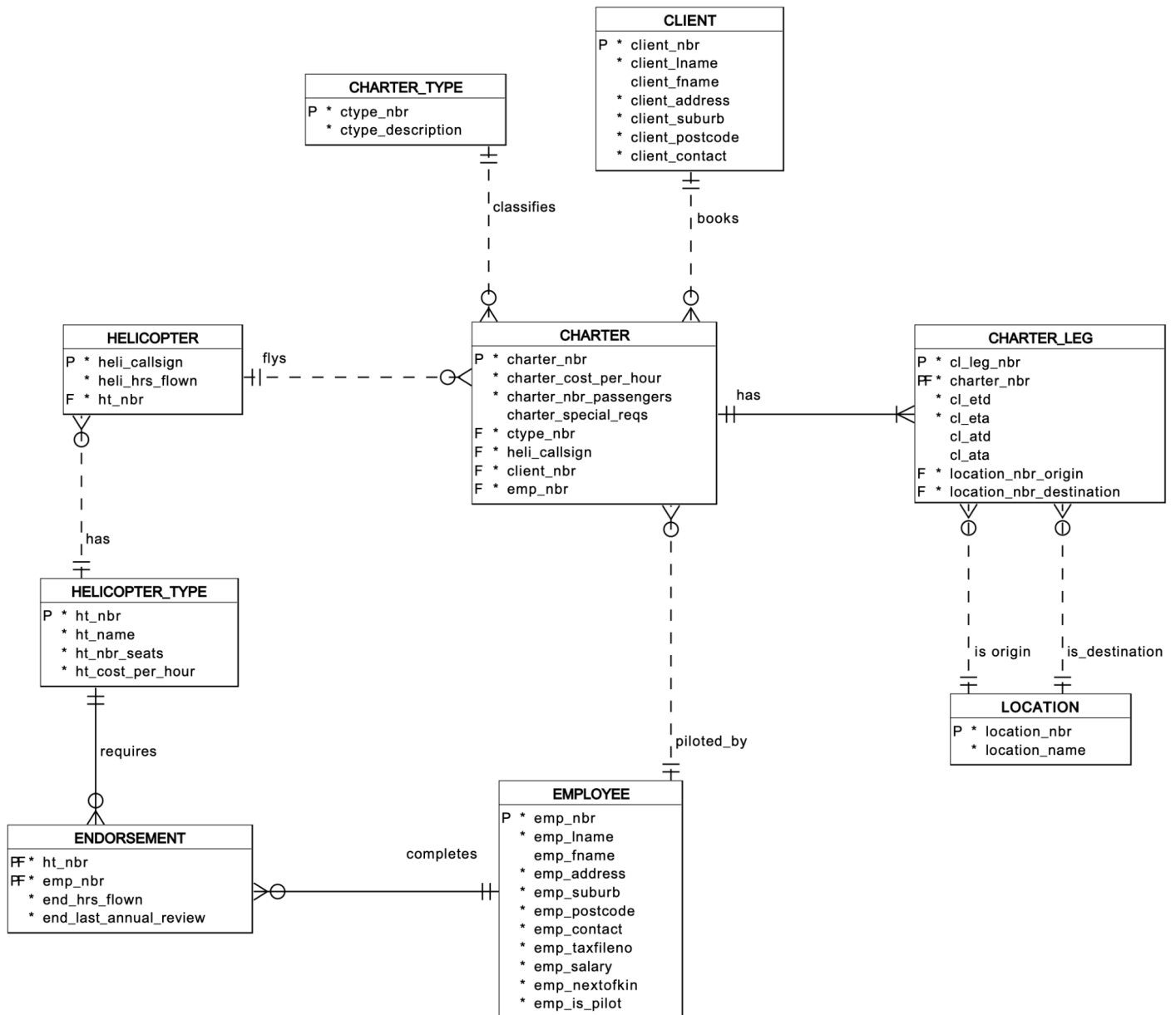
MonHeli is a thriving helicopter company. The company owns a range of helicopters whose standard cost of hire may vary depending on the type of charter and the number of hours flown. Each helicopter is identified by their call sign; much like a registration number identifies a car. The total time each helicopter is flown is recorded to keep track of maintenance schedules. The standard cost of hire for a helicopter depends on the helicopter type however this may be varied over time and for certain clients (eg. frequent flyers). Each of the helicopters has a fixed number of passenger seats available. The company's charter work is categorised as: scenic flight, transport, filming, geological survey, fire mapping, fuel reduction burning, logging coupe regeneration burning, aerial seeding, training, and search and rescue.

The company has a number of employees consisting of pilots, engineers and a secretary. Details are kept of each employee's name, address, contact number, Tax File Number, salary and next of kin. As only qualified pilots are allowed to take charters, records are kept of the pilots' endorsements. An endorsement is the credential a pilot has to fly a certain type of helicopter. Therefore, when a pilot is endorsed to fly a certain helicopter type (eg. JetRanger) a record is kept of the pilot's endorsement for this type of helicopter and the total hours the pilot has flown in this helicopter type. This indicates the pilot's level of experience in this type of aircraft. For a pilot to have a current endorsement for a helicopter type they must be reviewed annually and the date of their last review is kept for each of their endorsements.

Charters can vary in time – from several days to part of an hour. A charter is flown by one pilot using one helicopter. As clients only pay for the time spent flying, a record is kept of every time the helicopter takes off and lands during a charter flight (a leg) - a charter may consist of one or more legs. When a client initially makes a booking, the destination, the estimated time of departure (ETD) and estimated time of arrival (ETA) are recorded for each leg of the charter. For example, a client may book a helicopter to fly them from Gondor Airport to Rivendell for a 10:00 am meeting, then leave at 12:00 noon to fly to Mount Doom for 3 hours skiing and then back to Gondor Airport by 6:00 pm.

The actual time of departure (ATD) and actual time of arrival (ATA) are recorded for the charter so that clients are charged for the actual time spent flying (the accumulated engine time). The engine time for each leg is the difference in minutes between the ATA and ATD.

A data model has been developed for MonHeli, the logical model is shown below:



Task 1 - SQL Select (80 marks)

The tables for MonHeli have been created in the Monash Oracle server and are available from the user **MH** who has given you select rights on the tables. You will use these tables to answer the SQL Queries listed below.

Your answers for these tasks must be placed in the supplied SQL Script **mh-queries.sql**

You must ONLY use the data as provided in the text of the questions. Where a particular case (upper case, lower case, etc.) for a word is provided you **must only use that case**. You may divide names such as Zora Mandrey into the first name of Zora and a last name of Mandrey if required. **Failure to adhere to this requirement will result in a mark of 0 for the relevant question.**

ANSI joins must be used where two or more tables are to be joined, under no circumstances can "implicit join notation" be used - see the week 7 workshop slides and the week 8 tutorial.

You must NOT use PL/SQL or Views in Task 1.

Where a question indicates "Your output must have the form shown below" - this means the same appearance and alignment of columns/data as the sample output shows. Clearly your actual data may be different, this is a live database so changes may occur.

When required to show output which involves the client or pilot name as a full name, the name must not have any leading spaces (ie. not start with a space). *Where a full name is requested in any output it should be in the form first name last name eg. Legolas Woodland*

Q1

List all endorsements held by pilots where the endorsements last annual review date was after 31st March 2020.

Show the helicopter type number, employee number, employee lastname, employee firstname and the date of the review.

This listing should be displayed in ascending order of the last annual review date.

Your output must have the form shown below (partial output only shown):

HT_NBR	EMP_NBR	EMP_LNAME	EMP_FNAME	REVIEW_DATE
100	101	Gamgee	Samwise	Sun 31 May 2020
101	102	Brandybuck	Meriadoc	Sun 31 May 2020

[4 marks]

Q2

List those charters where special requirements are specified.

Show the charter number, client number, client lastname, client firstname and the special requirements specified.

This listing should be displayed in ascending order of the charter number.

[4 marks]

Q3

List the charter details for those charters which meet the following requirements:

- destination is Mount Doom (ie. charters with any leg that has a destination of mount doom should be included in the output), and
- the charter cost is less than \$1000 per hour, or no special requirements are specified

Show the charter number, client lastname, client firstname and the charter cost per hour. The client name should be listed in a single column called FULLNAME (sample output: Gandalf The Grey).

The listing should be displayed in the descending order of the client FULLNAME.

[8 marks]

Q4

For each helicopter type for which there are at least two helicopters of that type, list the total number of helicopters of that type.

Show the helicopter type number, helicopter type name and the number of helicopters owned.

The listing should be displayed in the descending order of the number of helicopters.

[8 marks]

Q5

For each location that has been used as the origin of a charter leg more than once, list the location as well as the number of times it has been used as an origin.

Show location number, location name and the number of times the location has been used as an origin.

The listing should be displayed in the ascending order of the number of times a location has been used as an origin.

[8 marks]

Q6

List the number of hours flown for EVERY helicopter type.

Show helicopter type number, helicopter type name and the total number of hours flown. Types which have no recorded hours should be shown as 0 hours flown.

The listing should be displayed in the ascending order of hours flown.

[8 marks]

Q7

List all the *completed* charters that Frodo Baggins has flown (completed flights), with the most recent charters appearing at the top of the list. *A completed charter means all legs have been flown.*

Show the charter number and the date/time of departure for leg one (note: a charter may span several days).

[8 marks]

Q8

List those *completed* charters where the total cost is less than the average total cost for all charters. The total cost for a charter is obtained by multiplying the charter cost per hour with the actual duration obtained from the leg actual departure and actual arrival times. *A completed charter means all legs have been flown.*

Show charter number, client number, client lastname, client firstname, total charter cost. The total charter cost should be rounded to two decimal digits and displayed with a \$ symbol e.g. \$1234.56

For this question, if either client name is empty '-' should be displayed.

The listing should be displayed in the descending order of total charter cost.

Your output must have the form shown below (partial output only shown):

CHARTER_NBR	CLIENT_NBR	CLIENT_LNAME	CLIENT_FNAME	TOTALCHARTERCOST
2	3	Woodland	Legolas	\$805.00
23	2	Elessar	Aragorn	\$758.33
30	3	Woodland	Legolas	\$758.33
19	1	The Grey	Gandalf	\$750.00
11	4	Gollum	-	\$700.00
18	2	Elessar	Aragorn	\$700.00

[10 marks]

Q9

Which charters have been able to depart at the time estimated for all legs of its flight?

Show the charter number, pilot's name and the client's name (both names should be shown in a single column).

Order the output by charter number.

Your output must have the form shown below (partial output only shown):

CHARTER_NBR	PILOTNAME	CLIENTNAME
3	Samwise Gamgee	Gandalf The Grey
4	Samwise Gamgee	Aragorn Elessar
5	Meriadoc Brandybuck	Legolas Woodland
6	Samwise Gamgee	Gollum

[10 marks]

Q10

For each client, list the name of their favorite destination location/s and how many times they have visited that destination. The clients favorite destination will be the location/s they visit the most based on completed legs.

Show the client number, client full name in one column, the name of the destination location and the number of times the location has been visited by the client.

Order the output by the client number. Where a client has several favourite destinations, order them by destination name.

[12 marks]

Task 2 - Triggers (20 marks)

Your answers for these tasks must be placed in the supplied SQL Script **mh-triggers.sql**

These tasks should be attempted only after task 1 has been successfully completed.

Use **mh-partial-schema.sql** to create and load data into HELI_TYPE, HELICOPTER, CHARTER, CHARTER_LEG, and LOCATION tables under your account. Note that these tables contain testing data only. Use these tables to answer the questions below.

For each of these questions, as part of your answer, you *must* create a set of SQL commands which will demonstrate the successful operation of your trigger (test harness) - *these tests are part of the awarded marks for each question*. Place these commands below your trigger definition for each of the tasks. Ensure your trigger definition finishes with a slash(/) followed by a blank line as detailed in the week 9 workshop and week 10 tutorial. In addition, when coding your trigger(s), you must provide output messages where appropriate.

Q1

MH, from now on, would like to maintain the integrity of the locations added into the CHARTER_LEG table. When the origin and destination locations are inserted/updated, the origin location must be different from the destination location. If both locations are the same, the trigger should prevent the action. Code a *single* trigger to enforce this requirement.

[4 marks]

Q2

MH, from now on, would like to automatically maintain the integrity of data inserted into the CHARTER table. When charter data is added or modified, the charter's cost per hour must be equal or higher than the helicopter's cost per hour. MH would also like to ensure that the charter's maximum number of passengers is not above the helicopter's number of seats. If either of these requirements is violated, then the trigger should prevent the action of inserting/updating the charter. Code a *single* trigger to enforce both of these requirements. Note: you may use any numeric values for ctype_nbr, client_nbr, and emp_nbr when testing your trigger.

[7 marks]

Q3

Create a trigger to automatically update the value of heli_hours_flown in the HELICOPTER table based on the actual time of departure and the actual time of arrival for a CHARTER_LEG.

The charter_leg data is inserted when the customer initially makes a booking with the estimated time of departure and arrival. The actual time of departure and arrival are updated when that leg is completed in a single update. Here is an example of the update statement:

```
UPDATE charter_leg
SET
    cl_atd = TO_DATE('01/06/2020 16:03', 'dd/mm/yyyy hh24:mi'),
    cl_ata = TO_DATE('01/06/2020 17:08', 'dd/mm/yyyy hh24:mi')
WHERE
    cl_leg_nbr = 2
    AND charter_nbr = 1;
```

The value of heli_hours_flown must then be updated based on the time the helicopter has actually flown for that charter leg.

Once the actual time of departure and arrival have been entered, they must not be allowed to be further updated.

[9 marks]

SUBMISSION REQUIREMENTS

Due Date: Friday 28th May at 5 PM AEST (week 12)

*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 two files you are **required** to submit:

- mh-queries.sql
- mh-triggers.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 a zip archive, submit the SQL script as it is. The SQL script must also have been pushed to the FIT GitLab server with an appropriate history as you developed your solutions (a minimum of four pushes). 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 SQL statements and constructs to:
 - retrieve the required data in the required format, and
 - where a layout or column heading has been specified, appropriately reflect these requirements,
- the correct application of triggers to maintain data integrity.

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

- contain SET ECHO ... or SPOOL commands,
- do not have a semicolon (;) closing the query for every query submitted,
- *do not make use of column aliases when you use arithmetic calculation, concatenation, functions, or other output manipulation unless specified otherwise in the above,*
- use subqueries and SQL conditions unnecessarily (although you are not required to consider efficiency of your solution you should try and ensure that you use the minimum number of subqueries and SQL conditions when arriving at your answer),
- use PL/SQL or Views in Task 1,
- *make use of WITH,*
- do not use to_char/to_date where appropriate in handling dates,
- do not include slash (/) followed by a blank line after the create trigger statements,
- do not provide output messages where appropriate when coding triggers,
- do not have an appropriate development history on the FIT GitLab server for all source files (**at least four pushes required**).