FIT9132 Introduction to Databases - Assignment 2B

Database - Monash New Smile

Purpose	Students will be asked to use SQL and NoSQL to write queries to produce specified output. This task covers learning outcomes: 1. Apply the theories of the relational database model. 4. Manage data that meets user requirements, including queries and transactions. 5. Contrast the differences between non-relational database models and the relational database model.
Your task	This is an open book, **individual task**. The final output for this task will be a set of relational (RA and Oracle) and non relational (MongoDB) queries which meet the user requirements.
Value	**35%** of your total marks for the unit
Due Date	Monday, **30th October 2023**, 23:55
Submission	● Via Moodle Assignment Submission ● FIT GitLab check ins will be used to assess history of development
Assessment Criteria	● Application of relational algebra operations to produce outputs that meet user requirements ● Application of SQL select statements to produce outputs that meet user requirements. ● Mapping of relational database data into non relational database data structure. ● Application of MongoDB operations to produce outputs that meet user requirements.
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

Monash New Smile (MNS) provides dental services such as fillings, scalings, extractions etc to its patients. For each service, MNS records a service code, service description and the MNS standard fee for this service. The actual fee charged to a patient for a particular service may be varied from this standard fee. These services are provided by a range of professionals (providers) that MNS employs. Each provider is assigned a provider code. MNS records the provider's title, name, the specialisation, and the room number where they normally treat patients. Each provider may have a particular specialisation (one only), but some providers may not have any specialisation. Some services require a particular provider while other services are able to be provided by a number of providers.

MNS patients are assigned a patient number. The company records the patient's name, date of birth, the residential address, contact phone number and contact email address (all patients are required to provide a contact number (mobile phone) and an email address for the purpose of confirming appointments). Patients must also provide another person's name and phone number for emergency purposes, which is also recorded.

Patients contact Monash New Smile and make appointments to see a provider. Each appointment is booked with only a single provider, the provider is assigned when the appointment is first recorded. The system also ensures that patients, providers, and rooms are not double booked (e.g. a patient cannot have two scheduled appointments at the same date time). There is a possibility that a patient needs more than one appointment per day (e.g. having an X-Ray procedure in the morning and coming back in the afternoon for re-evaluation). An appointment can be a follow up of a prior appointment. MNS records the prior appointment number for each follow up appointment. For example, appointment number 4 is a follow up of appointment number 1, and appointment number 31 is a follow up of appointment 4.

An appointment will require one or more services depending on the work required. For an appointment, MNS records the appointment date and time and the room number in which the consultation (appointment) will take place - in some circumstances this room may not be the provider's normal room, for example, due to the need to use specialist equipment. The reception staff, based on the patient's requirements, set an appointment length as either short (30 minutes), standard (60 minutes) or long (2 hours). The staff who schedule appointments will also ensure that the allocated provider is available and can provide the services which will be required during this appointment, your design is not required to enforce this, although these staff must be able to look up which providers provide which services if required.

Providers require the assistance of one dental nurse during an appointment. Each nurse is assigned a nurse number - MNS also records the nurse's name and their contact number. One nurse is assigned to each appointment. Nurses employed by MNS are hired based on three employment status: Casual (C), Contract (T), and Fulltime (F).

Each service provided at a particular appointment may require several items, for example fillings require Porcelain Etch and Silane. MNS records the quantity of items needed in each service provided at a particular appointment. Each item is assigned a unique identifier. MNS also records

the item's description, standard cost and the number of items that are currently on hand. This figure is needed to make decisions about placing item purchase orders.

Based on these requirements a data model has been created for MNS:
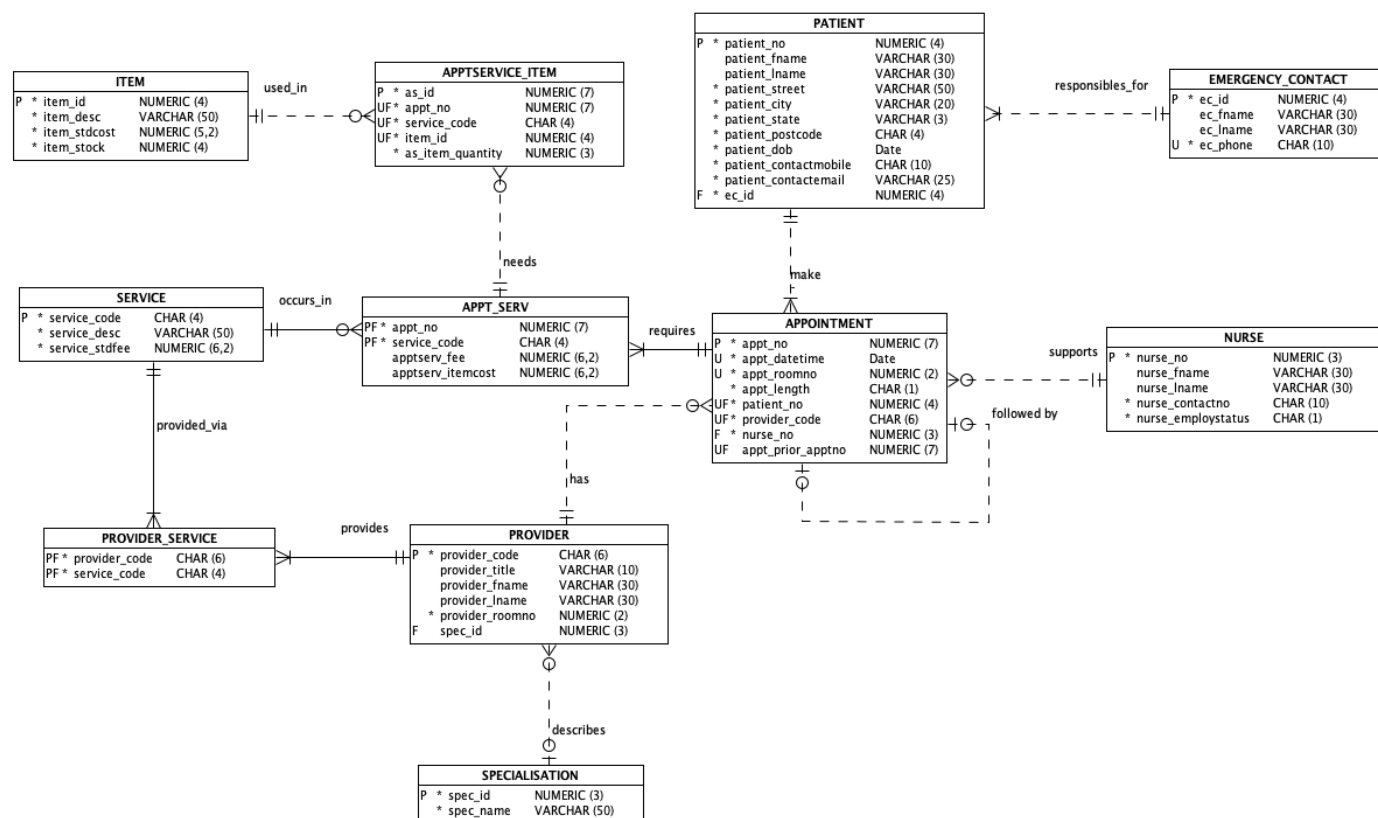
TASKS

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 <u>individual local working directory</u> (repo) in the Assignment 2B folder and **regularly pushed to the FIT GitLab server to build a clear history of development of your approach**. A minimum of twelve pushes to the FIT GitLab server is required (3 pushes per file). Please note twelve pushes is a *minimum*, in practice we would expect significantly more. All commits must include a **meaningful commit message** which clearly describes what the particular commit is about and **must be correctly assigned to your valid GitLab author**.

You must regularly check that your pushes have been successful by logging in to the web interface of the FIT GitLab server; you must not simply *assume* they are working. Before submission, via Moodle, you **must** log in to the [web interface of the GitLab server](#) and ensure your submission files are present on the GitLab server and that their names are unchanged.

TASK 1: Relational Database Queries - Relational Algebra (9 marks):

Your answers for this task (Task 1) must be written in an Ms Word document or Google document. Once you have completed all questions, download or print the document as a pdf file and name the file as **T1-mns-ra.pdf**.

For this task you are required to write the **relational algebra operations** for the following queries (your answer must show an understanding of query efficiency).

Example:

Question:
List the service code and service description for all services which have a standard fee greater than $200.

Answer:

$R = \pi_{service\_code, service\_desc} (\sigma_{service\_stdfee>200} SERVICE)$

OR

$R1 = \sigma_{service\_stdfee>200} SERVICE$
$R = \pi_{service\_code, service\_desc} (R1)$

List of symbols for copying/pasting as you enter your answers below:

project: π, select: σ, join: ⋈, intersect: ∩, union: ∪, minus: -

(a) List the id and description of all items which have never been used in any appointment service.

[2 marks]

(b) List the patient number, patient first name, patient last name, emergency contact first name, emergency contact last name and emergency contact phone number of all patients who live in a city named *Mooroolbark* and had appointment/s on *08 September 2023*.

[3 marks]

(c) List the number, first name, last name and email address of all patients who have been attended by endodontists (i.e. providers who specialise in *ENDODONTICS*).

[4 marks]

TASK 2: Relational Database Queries - SQL (63 marks):

****

Before attempting this task, **drop all tables under your account**.

In arriving at your solutions for Task 2 you are ONLY permitted to use the SQL structures, syntax and functions **which have been covered within this unit**:

- Week 8 Workshop and Week 9 Applied - SQL Part I - Basic
- Week 9 Workshop and Week 10 Applied - SQL Part II- Intermediate
- Week 10 Workshop and Week 11 Applied - SQL Part III - Advanced

SQL syntax and commands outside of the covered work, as detailed above, **will NOT be accepted or marked**.

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. *ALL necessary calculations must be carried out as part of your SQL code*,
- assume any contents in the database - rows in a table are potentially in a constant state of change, or
- use views and/or PL/SQL

Your answers must recognise the fact that you have been given 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 <u>multiple users will be operating in the tables at the same time</u>. 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.**

****

In the Monash Oracle database, this Monash New Smile set of tables has been created under the user "MNS". To use these tables you need to add the prefix "MNS" to the table names that you use in an SQL statement. So, if you want to retrieve data from PROVIDER table you need to write:

```
select * from mns.provider;
```

Your answers for this task (Task 2) must be placed in the supplied SQL script **T2-mns-select.sql.**

You are only permitted to code **a single select statement for each question** below. Where you have been supplied with a string contained in *italics*, such as *Orthodontics* you must search in the database using the provided string exactly as supplied. Where you need to show a full name, you must not have any extra spaces (e.g. leading space, extra space in the middle of the name).

Sample output showing the **form** of what you are required to produce is provided. Note this *is the form of the output ONLY i.e. the appearance*, the data you return will be different.

(a) List the id, description, standard cost and stock of all items which have a stock of at least 50 items (a minimum of 50) and include the word *composite* in their item description. Show the items with the highest stock first. If two items have the same number of items on hand (stock), order them by their respective item id.

[5 marks]

(b) List the provider code and provider name in the form of Title. FirstName LastName (e.g. Dr. Bruce Striplin) for all providers who specialise in the area of *PAEDIATRIC DENTISTRY* (this is the specialisation description). Order the list by the providers' last name and within this, if two providers have the same last name, order them by their respective first names, then by their provider codes.

[6 marks]

(c) List the service code, description, and standard fee where the service is more expensive than the average standard fee for all services. The output must show the most expensive service first. The service standard fee must be right aligned and displayed with two decimal digits and a leading $ symbol, for example as $250.50. Where two or more services have the same standard fee, order the output by service code.

[8 marks]

(d) List the appointment number, appointment date time, patient number, patient full name and appointment total cost (service fee/s + item fee/s) for the most expensive appointment/s. The appointment total cost must be right aligned and displayed with two decimal digits and a leading $ symbol, for example as $2575.50. Where two or more appointments have the same appointment total cost, order the output by appointment number.

[8 marks]

(e) Given a provider may charge more or less than the standard fee for a service carried out during an appointment, the MNS administration is interested in finding out what variations on the fee have been charged across all appointments. The clinic terms the difference between the *average* actual charged service fee and the service standard fee as the "Service Fee Differential". For all services which have been carried out during an appointment, determine the service fee differential. The list should show the service code, description, standard fee and the service fee differential in service code order.

For example service D001 Oral Examination has a standard fee of $65.00, it may have been charged on average across all appointments for $75.00 - the fee differential here will be 75 - 65 that is a fee differential +10.00. If the average charge had been say 57.50 the fee differential will be -7.50.

[10 marks]

(f) List for all patients the percentage of their appointments which are follow up appointments. For example if a patient has made three appointments, of which two were followup appointments then this percentage will be 66.7%. Display the patient number, full name, current age in years, number of appointments which have been made and the percentage of those appointments which were followup appointments. The image below shows the output you are required to produce, you must use the same column headings, data formats and alignment.

```
PATIENT_NO PATIENTNAME             ... CURRENTAGE   NUMAPPTS FOLLOWUPS
---------- ----------------------- ... ---------- ---------- ---------
         1 Gemma Hansman           ...         21          4     75.0%
         2 Ashley Soares           ...         24          2      0.0%
         3 Marie-Rose Johnson      ...         56          1      0.0%
         4 Darcy                   ...         13          2     50.0%
         5 Sally Johnson           ...         10          1      0.0%
```

Image note:
 - only sample/incomplete data shown (yours will be different)
 - the PATIENTNAME column has been truncated for easier display
 - … represents the part of the column removed/truncated

[11 marks]

(g) List for ALL providers the number of appointments they have had, the total fees they have charged and the total quantity of all items that they have used for appointments which took place between 9 AM on the 10th September 2023 and 5 PM on the 14th September 2023 (inclusive). If a particular provider has had no appointments in this period, or did not collect fees, or used no items the output columns must show a dash (-) The image below shows the output you are required to produce, you must use the same column headings, data formats and alignment (all output is right aligned).

```
 PCODE  NUMBERAPPTS  TOTALFEES  NOITEMS
------- ------------ ---------- -------
AST001            4 $22,400.00        7
AST002            1  $1,035.00        -
GEN001            -          -        -
```

Image note:
 - only sample/incomplete data shown (yours will be different)

[15 marks]

TASK 3: Non Relational Database Queries - MongoDB (18 marks):

Your answers for this task (Task 3) must be placed in the supplied sql file **T3-mns-json.sql** and the supplied MongoDB script file **T3-mns-mongo.mongodb.js**

(a) Write an SQL statement in **T3-mns-json.sql** to generate a collection of JSON documents using the following structure/format from the **MNS** tables. Each document in the collection represents an appointment's provider and item details. The list must only include appointments which used at least one item. Note that the _id in this structure is the appointment number.

```
{
    "_id": 7,
    "datetime": "08/09/2023 12:00",
    "provider_code": "GEN002",
    "provider_name": "Dr. Amalia Morris",
    "item_totalcost": 78,
    "no_of_items": 1,
    "items": [
        {
            "id": 18,
            "desc": "Fluid composite",
            "standardcost": 78,
            "quantity": 1
        }
    ]
},
…
{
    "_id": 11,
    "datetime": "11/09/2023 15:00",
    "provider_code": "PED002",
    "provider_name": "Dr. Lee",
    "item_totalcost": 80,
    "no_of_items": 2,
    "items": [
        {
            "id": 20,
            "desc": "Phospor imaging plate",
            "standardcost": 75,
            "quantity": 1
        },
        {
            "id": 21,
            "desc": "Clinasept Film",
            "standardcost": 5,
            "quantity": 1
        }
    ]
}
… //partial collection only
```

[7 marks]

Write the MongoDB commands for the following questions, 3(b) - 3(e), in the supplied MongoDB script file named **T3-mns-mongo.mongodb.js**.

(b) Create a new collection and insert all documents generated in 3(a) above into MongoDB. Provide a drop collection statement right above the create collection statement. After the documents have been inserted, use an appropriate db.find command to list all the documents you added.

[1 mark]

(c) List all appointments which use more than two different items or their total item cost are more than $50.

[2 marks]

(d) The management decided to change the description of item id 1 from 'Paper tips' into 'Paper points'. Change the data within the collection to reflect this decision.

Use an appropriate db.find command after making the change so that you illustrate/confirm the change which was made.

[3 marks]

(e) Insert three additional items into appointment number 20 (i.e. _id is 20):

- 'EDTA Cleansing Gel 17%' (item id: 3, standard cost: $8) x 1 used
- 'Irrigation Solution 2% Chlorhexidine' (item id: 4, standard cost: $9) x 1 used, and
- 'Irrigation Needle and Syringe' (item id: 8, standard cost: $2) x 2 used.

Write the necessary MongoDB commands to add these new items into the collection.

Use an appropriate db.find command after making the change so that you illustrate/confirm the change which was made.

[5 marks]

Submission Requirements

Due Date: **30th October 2023 at 23:55**

*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 to Moodle and must exist in your individual FIT Gitlab repository **under Ass2B folder**:

- T1-mns-ra.pdf
- T2-mns-select.sql
- T3-mns-json.sql
- T3-mns-mongo.mongodb.js

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 files. 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 12 pushes - 3 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 solutions will be assessed against an optimal solution for this task - this optimal solution will be available as a sample solution after Assignment 2B has been graded.

TASK	Items Assessed
TASK 1 Relational Algebra Queries 9 marks	
	Maximum 9 marks - Satisfy brief requirements: ● Marks awarded, as listed, (a) - (c) for relational algebra which meets the expressed requirement ● Marks awarded for correct use of RA operations and the use of appropriate symbols ● Mark penalty applied if your answer does not show an understanding of query efficiency i.e. you must not make use of unnecessary joins, nor carry attributes and tuples up through the query which are not necessary ● 0 marks will be awarded if SQL select statements are provided as solutions for task 1
TASK 2 Relational Database Queries 63 marks	
	Maximum 63 marks - Satisfy brief requirements: ● Marks awarded, as listed, (a) - (f) for SQL code which meets the expressed requirement ● Mark penalty applied if output does not match the form supplied for each question ● Mark penalty applied if date handling and string database lookups are not managed correctly ● Mark penalty applied if column aliases are not used when arithmetic calculation, concatenation, functions, or other output manipulation is used in a query ● Mark penalty applied if manual lookup/calculation is used for values from the database ● Statements which do not execute correctly in Oracle will be awarded a maximum of 50% of the available marks less 1 mark. For example, if a question is worth 6 marks and runs with an error in SQL the *maximum* mark awarded will be 2 marks
Task 3 Non Relational Database Queries - MongoDB 18 marks	
	Maximum 18 marks - Satisfy brief requirements: ● Maximum of 5 marks awarded for creation of a JSON document which matches the supplied document format ● Marks awarded, as listed, (b) - (e) for MongoDB code which meets the expressed requirement ● Mark penalty applied if field names and predicates (such as "&eq") are not enclosed in double quotes

	● Statements which do not execute correctly in MongoDB will be awarded a maximum of 50% of the available marks less 1 mark. For example, if a question is worth 6 marks and runs with an error in MongoDB the *maximum* mark awarded will be 2 marks
Correct use of Git 10 marks	
	Maximum 10 marks - Git used appropriately: ● Marks awarded for twelve pushes showing a clear development history for the files (minimum three pushes per file) ● Marks awarded for correct Git author details used in pushes (see week 2 Applied notes Appendix) ● Marks awarded for the use of meaningful commit messages (i.e. not blank or of the form "Push1")
Penalties	
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 Task 2, will result in a ***grade deduction of 10 marks being applied.***
Use of SQL syntax in Task 2 which has not been covered in this unit	Such work will **NOT BE GRADED** (see page 5 of this brief)

Final Assignment Mark Calculation

Total:100 marks, recorded as a grade out of 35