# Skills assessment: Create and query a database

## Criteria

### Unit code and name

ICTDBS416 | Create basic relational databases

ICTPRG431 | Apply query language in relational databases

ICTSAD501 | Model data objects

### Qualification/Course code and name

ICT40120 | Certificate IV in Information Technology

## Student details

### Student number

881299424

### Student name

Ramazan Carpan

## Assessment declaration

*Note: If you are an online student, you will be required to complete this declaration on the TAFE NSW online learning platform when you upload your assessment.*

This assessment is my original work and has not been:

* plagiarised or copied from any source without providing due acknowledgement.
* written for me by any other person except where such collaboration has been authorised by the Teacher/Assessor concerned.

### Student signature and date

Ramazan.Carpan 10/04/2024

Version: 20210920

Date created: 20 September 2021

Date modified: 20 September 2021

For queries, please contact:

Technology and Business Services Skills Point

Ultimo

© 2021 TAFE NSW  
RTO Provider Number 90003 | CRICOS Provider Code: 00591E

This assessment can be found in the: [Learning Bank](https://share.tafensw.edu.au/share/access/searching.do?doc=%3Cxml%2F%3E&in=P7ac4831b-430a-4b8d-8b56-f7b32ed5b9cf&q=&type=standard&sort=rank&dr=AFTER)

The contents in this document is copyright © TAFE NSW 2021 and should not be reproduced without the permission of TAFE NSW. Information contained in this document is correct at the time of printing: 8 April 2024. For current information please refer to our website or your Teacher/Assessor as appropriate.

## Assessment instructions

Table 1 Assessment instructions

| Assessment details | Instructions |
| --- | --- |
| **Assessment overview** | The aim of this assessment is to assess your skills in creating a database and retrieving and manipulating information using a query language. |
| **Assessment event number** | 3 of 4 |
| **Instructions for this assessment** | This is a skills-based assessment that assesses your ability to demonstrate skills required in the unit.  This assessment is in three parts:   1. Create the database 2. Query the database 3. Use views and stored procedures   And is supported by:   * Assessment feedback * Supporting documents (within the assessment)   **Note**: This assessment may contain links to external resources. If a link does not work, copy and paste the URL directly into your browser. |
| **Submission instructions** | On completion of this assessment, you are required to submit it to your Teacher/Assessor for marking. Where possible, submission and upload of all required assessment files should be via the TAFE NSW online learning platform.  It is important that you keep a copy of all electronic and hardcopy assessments submitted to TAFE and complete the assessment declaration when submitting the assessment. |
| **What do I need to do to achieve a satisfactory result?** | To achieve a satisfactory result for this assessment you must be available at the arranged time to complete all the assessment criteria as outlined in the assessment instructions.  All parts of the observable task must be performed to a satisfactory level as indicated in the criteria section of the observation checklist.  All oral questions must be answered correctly to be deemed satisfactory in this assessment task; however, Teachers/Assessors may ask you additional questions to confirm your understanding of the task.  If a resit is required to achieve a satisfactory result it will be conducted at an agreed time after a suitable revision period. |
| **What do I need to provide?** | * TAFE NSW student account username and password. If you do not know your username and password, contact your campus or service centre on 131601. |
| **What the Teacher/Assessor will provide** | * Access to simulated workplace environment * Software development environment and resources * Requirement documentation * Data source * Entity relationship diagram * Calculation resources, such as Microsoft Excel |
| **Due date**  **Time allowed**  **Venue** | Refer to UAG  Indicative time to complete assessment:   * In class: Three hours * Out of class: Two hours.   TAFE NSW campus/ TAFE Digital Campus/ TAFE NSW Moodle/ a location determined by your Teacher/Assessor |
| **Assessment feedback, review or appeals** | In accordance with the TAFE NSW policy *Manage Assessment Appeals,* all students have the right to appeal an assessment decision in relation to how the assessment was conducted and the outcome of the assessment. Appeals must be lodged within **14 working days** of the formal notification of the result of the assessment.  If you would like to request a review of your results or if you have any concerns about your results, contact your Teacher/Assessor or Head Teacher. If they are unavailable, contact the Student Administration Officer.  Contact your Head Teacher/Assessor for the assessment appeals procedures at your college/campus. |

## Specific task instructions

The instructions and the criteria in the tasks and activities below will be used by the Teacher/Assessor to determine whether the tasks and activities have been satisfactorily completed. Use these instructions and criteria to ensure you demonstrate the required skills and knowledge.

### Scenario

You will use the medical practice data model indicated below for this assessment. The data model for this assessment consists of the ***Medical practice database data dictionary*** and the ***Medical practice database crow’s foot entity relationship diagram (ERD)***.

#### Medical practice database data dictionary

| Table name | Attribute name | Contents | Data type | Format | Range | Reqd | Key type | FK referenced table |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Patient | Patient\_ID | Patient's unique system generated identifier | INTEGER | 99999 | 10000-99999 | Y | PK |  |
|  | Title | Patient's title, e.g. Mr, Ms, Mrs, Dr, etc. | NVARCHAR(20) | Xxxxxx |  |  |  |  |
|  | FirstName | Patient's first name | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | MiddleInitial | Patient's middle initial of their name | NCHAR(1) | X |  |  |  |  |
|  | LastName | Patient's last name | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | HouseUnitLotNum | The number of Patient's residence, unit, or lot | NVARCHAR(5) | XXXXX |  | Y |  |  |
|  | Street | Name of street, road, etc. where patient resides | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | Suburb | Name of suburb where patient resides | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | State | Name of state where patient resides | NVARCHAR(3) | XXX |  | Y |  |  |
|  | PostCode | Post Code of Patient's residential address | NCHAR(4) | XXXX |  | Y |  |  |
|  | HomePhone | Patient's home phone number | NCHAR(10) | XXXXXXXXXX |  |  |  |  |
|  | MobilePhone | Patient's mobile phone number | NCHAR(10) | XXXXXXXXXX |  |  |  |  |
|  | MedicareNumber | Patient's Medicare number | NCHAR(16) | XXXXXXXXXXXXXXXX |  |  |  |  |
|  | DateOfBirth | Patient's date of birth | DATE | dd/mm/yyyy |  | Y |  |  |
|  | Gender | Patient's gender | NCHAR(20) | male, female, unspecified, indeterminate, intersex |  | Y |  |  |
| Practitioner | Practitioner\_ID | Practitioner's unique system generated identifier | INTEGER | 99999 | 10000-99999 | Y | PK |  |
|  | MedicalRegistrationNumber | Practitioner's Medical Registration Number | NCHAR(11) | XXXXXXXXXXXXXXXX |  | Y | AK |  |
|  | Title | Practitioner's title, e.g. Mr, Ms, Mrs, Dr, etc. | NVARCHAR(20) | Xxxxxx |  |  |  |  |
|  | FirstName | Practitioner's first name | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | MiddleInitial | Practitioner's middle initial of their name | NCHAR(1) | X |  |  |  |  |
|  | LastName | Practitioner's last name | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | HouseUnitLotNum | The number of Practitioner's residence, unit, or lot | NVARCHAR(5) | XXXXX |  | Y |  |  |
|  | Street | Name of street, road, etc. where patient resides | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | Suburb | Name of suburb where patient resides | NVARCHAR(50) | Xxxxxx |  | Y |  |  |
|  | State | Name of state where patient resides | NVARCHAR(3) | XXX |  | Y |  |  |
|  | PostCode | Post Code of Practitioner's residential address | NCHAR(4) | XXXX |  | Y |  |  |
|  | HomePhone | Practitioner's home phone number | NCHAR(8) | XXXXXXXXXX |  |  |  |  |
|  | MobilePhone | Practitioner's mobile phone number | NCHAR(8) | XXXXXXXXXX |  |  |  |  |
|  | MedicareNumber | Practitioner's Medicare number | NCHAR(16) | XXXXXXXXXXXXXXXX |  |  |  |  |
|  | DateOfBirth | Practitioner's date of birth | DATE | dd/mm/yyyy |  | Y |  |  |
|  | Gender | Practitioner's gender | NCHAR(20) | male, female, unspecified, indeterminate, intersex |  | Y |  |  |
|  | PractitionerType\_Ref | The type of Practitioner | NVARCHAR(50) | XXXXXXXXXXXXXXXX |  | Y | FK | PractitionerType |
| Availability | WeekDayName\_Ref | The name of the day of the week | NVARCHAR(9) |  | Mon-to-Fri | Y | PK, FK | WeekDays |
|  | Practitioner\_Ref | The ID of the Practitioner | INTEGER | 99999 | 10000-99999 | Y | PK, FK | Practitioner |
| WeekDays | WeekDayName | The name of the day of the week | NVARCHAR(9) | Xxxxxxxxxx | Mon-to-Fri | Y | PK |  |
| PractitionerType | PractitionerType | The type of Practitioner | NVARCHAR(50) | Xxxxxxxxxx |  | Y | PK |  |
| Appointment | Practitioner\_Ref | The ID of the Practitioner | INTEGER | 99999 | 10000-99999 | Y | PK, FK | Practitioner |
|  | AppDate | The date of the appointment | DATE | dd/mm/yyyy |  | Y | PK, AK |  |
|  | AppStartTime | The time of the appointment | TIME | 00:00 AM/PM |  | Y | PK, AK |  |
|  | Patient\_Ref | The ID of the Patient | INTEGER | 99999 | 10000-99999 | Y | FK, AK | Patient |

##### Legend

|  |  |
| --- | --- |
| Abbreviation | Description |
| PK | Primary key |
| FK | Foreign key |
| AK | Alternate key |
| NCHAR | Fixed length Unicode Character Data |
| NVARCHAR | Variable length Unicode Character Data |

#### Medical practice database crow’s foot ERD

Medical practice database crow’s foot ERD
Tables:
Patient
Appointment
Practitioner
PractitionerType
Availability
WeekDays

## Part 1: Create the database

To complete this part of the assessment, you are required to complete the practical tasks described below. These tasks require you to develop Structured Query Language (SQL) code that addresses the requirements of each task.

These practicals may be observed by your assessor. The code developed by you in each task will be used by your assessor to assess you, and you must submit your code as evidence for this assessment.

Download and unzip the [resource folder](https://share.tafensw.edu.au/share/items/e08936de-a951-4b19-be3f-11ccf6bd2d31/0/?attachment.uuid=98bd5196-6b67-4b42-b29b-76add0c8ecd4) (Cl\_Database\_AE\_Sk\_Appx.zip) to access files you will need for this part.

**Note: You will create one script file for Part 1 Task 1 below, then append to the same file the SQL script for tasks 2 and 5.**

1. Develop a script using Data Definition Language (DDL) elements of Structured Query Language (SQL) to create a database with the name MedicalPractice. Save your script file as yourFirstname\_yourSurname\_CreateMedicalPracticeDatabase.sql.
2. Using DDL elements of SQL, write the script to create each of the tables described in the medical practice data model, appending to the end of the script file. Your script must precisely match the details specified in the data model and must include creation of the:
   1. tables, using the table names specified by the data model
   2. columns, using the column names specified by the data model
   3. column data types using the data types and sizes specified by the data model
   4. primary key constraints specified by the data model
   5. foreign key constraints required to implement the relationships specified by the data model
   6. unique index constraints required to implement the alternate keys specified by the data model. Note the following alternate keys exist in the data model:

|  |  |
| --- | --- |
| Table | Alternate Key |
| Practitioner | MedicalRegistrationNumber |
| Appointment | Patient\_Ref, AppDate, AppStartTime |

1. Execute your script on the Database Server. Check and ensure that your script executes without errors and that it creates the database, tables and constraints as defined by the Medical Practice Data Model above. If the database has been created successfully, you can now proceed to the next task and populate the database with data. Otherwise, correct any errors in your script, before you proceed.
2. Test the database.
   1. Create a test plan using the [Software Test Report template](https://share.tafensw.edu.au/share/items/02285ff1-cfb2-4af4-b402-fdc23bf4bf11/0/?attachment.uuid=074bd9cc-f9e2-46fc-81ad-a43cc32c218e) (GE\_Software-Test-Report\_template.docx) to test the database to ensure it matches the requirements. Include:
      * sample test data
      * expected outcomes.
   2. Test and debug the database using your test plan and record the test results:
      * Correct any errors and include screenshots.
      * Write an outline of the errors that you corrected.
3. Write and execute the script to populate the database tables using the supplied data files:
   1. AppointmentData.csv
   2. AvailabilityData.csv
   3. PatientData.csv
   4. PractitionerData.csv
   5. PractitionerTypeData.csv
   6. WeekDaysData.csv.
4. When you have completed Tasks 1-5:
   1. confirm with your assessor that you have completed this part of the assessment
   2. ask for feedback and make any changes as required
   3. submit your Software Test Report to your assessor
   4. ensure your assessor signs your completed Software Test Report. This may be printed and signed, an electronic signature or a reply email with sign off.

**Submit the following for Part 1:**

* SQL file containing scripts for tasks 1, 2 and 5
* Completed and signed Software Test Report.

## Part 2: Query the database

To complete this part of the assessment, you are required to complete the practical tasks described below. These tasks require you to develop SQL code that addresses the requirements of each task.

These practicals may be observed by your assessor. The code developed by you in each task will be used by your assessor to assess you, and you must submit your code as evidence for this assessment.

### Task 1

In Task 2, you will be building database queries. Based on recommendations from your teacher/assessor, confirm the necessary tools and environment/database platform that you are required to use to **query** the database.

Table 2: Tools and environment

|  |  |
| --- | --- |
| Type | Tool/environment/platform |
| Operating System | Windows |
| Relational Database Management System | SQL Server |
| Query tool | Azure Data Studio |

### Task 2

Write a SQL query to provide the information requested for each of the following requirements.

To check your results, you can make estimations of your answer using a suitable tool, such as a spreadsheet or calculator and compare your estimate to the answer from your query.

**Note: You will create one script file that includes all the queries for Task 2.** Save your script file as *yourFirstname\_yourSurname\_Queries.sql*.

1. List the first name and last name of female patients who live in St Kilda or Lidcombe.
2. List the first name, last name, state and Medicare Number of any patients who do not live in NSW.
3. List each patient's first name, last name, Medicare Number and their date of birth. Sort the list by date of birth, listing the youngest patients first.
4. For each practitioner, list their ID, first name, last name, the total number of days and the total number of hours they are scheduled to work in a standard week at the Medical Practice. Assume a workday is nine hours long.
5. List the Patient's first name, last name and the appointment date and time, for all appointments held on the 18/09/2019 by Dr Anne Funsworth.
6. List the ID and date of birth of any patient who has not had an appointment and was born before 1950.
7. List the patient ID, first name, last name and the number of appointments for patients who have had at least three appointments. List the patients in 'number of appointments' order from most to least.
8. List the first name, last name, gender, and the number of days since the last appointment of each patient and the 23/09/2019.
9. List the full name and full address of each practitioner in the following format exactly.  
   Dr Mark P. Huston. 21 Fuller Street SUNSHINE, NSW 2343  
   Make sure you include the punctuation and the suburb in upper case.

Sort the list by last name, then first name, then middle initial.

1. List the patient id, first name, last name and date of birth of the fifth oldest patient(s).
2. List the patient ID, first name, last name, appointment date (in the format 'Tuesday 17 September, 2019') and appointment time (in the format '14:15 PM') for all patients who have had appointments on any Tuesday after 10:00 AM.
3. Create an address list for a special newsletter to all patients and practitioners. The mailing list should contain all relevant address fields for each household. Note that each household should only receive one newsletter.

**Submit the following for Part 2:**

* This assessment document with Task 1 completed
* SQL file containing scripts for Task 2.

## Part 3: Use views and stored procedures

To complete this part of the assessment, you are required to complete the practical tasks described below. These tasks require you to develop SQL code that addresses the requirements of each task.

These practicals may be observed by your assessor. The code developed by you in each task will be used by your assessor to assess you, and you must submit your code as evidence for this assessment.

**Note: You will create one script file that includes all the queries for Part 3.** Save your script file as *yourFirstname\_yourSurname\_ViewsSP.sql*.

### Tasks

1. Create a **view** (called *vwNurseDays*) with the name and phone details of any nurse (registered or not) and the days that they work. Execute the SQL statements to create the view.
2. Using your **view**, write a query to retrieve the name and phone number details of all nurses who are scheduled to work on a Wednesday.
3. Create a **view** (called *vwNSWPatients*) that contains all patient details for patients whose address is in NSW. Execute the SQL statements to create the view.
4. Create a **stored procedure** (called *spSelect\_vwNSWPatients*) to retrieve all records and columns from *vwNSWPatients* in postcode order ascending*.* Execute the stored procedure.
5. Create a **stored procedure** (called *spInsert\_vwNSWPatients*) to insert a new record into *vwNSWPatients*, using parameters for all relevant data. Execute the stored procedure inserting a record for a new patient named Mr Mickey M Mouse from 1 Smith St, Smithville, NSW 2222.

Run a query to verify that the record has been added to the *Patient* table.

1. Create a **stored procedure** (called *spModify\_PractitionerMobilePhone*) using the Practitioner table to change a practitioner’s mobile phone number, using the Practitioner ID and the new mobile number as parameters. Execute the stored procedure to change Hilda Brown’s mobile number to 0412345678.

Run a query to verify that the record has been updated in the *Practitioner* table.

1. Manipulate the Practitioner table to store a driver’s licence number. For privacy and security purposes this data needs to be encrypted. Name the new column DriversLicenceHash. For encrypting the column, use a one-way hashing algorithm. Execute the statement to create the new column.

Add the drivers licence number of 1066AD Dr Ludo Vergenargen’s (Practitioner ID 10005) drivers licence using a SHA hashing function.

Display Dr Vertenargen’s record to view the hashed number.

1. Manipulate the *Patient* table to add a new column that will store a date value which is the last date they made contact. The default value should be the date of record creation. Name the new column *LastContactDate*. Execute the statement to create the new column.
2. Create a **trigger** on the *Appointment* table that will update *LastContactDate* on the Patient table each time a new record is added to the Appointment table. The value of the *LastContactDate* should be the date the record is added. Name the trigger *tr\_Appointment\_AfterInsert*.
3. Delete the view *vwNurseDays* from the database.
4. Delete the stored procedure *spSelect\_vwNSWPatients* from the database.

**Submit the following for Part 3:**

* SQL file containing scripts for Task 3.

## Assessment feedback

*NOTE: This section must have the Teacher/Assessor and student signature to complete the feedback. If you are submitting through the TAFE NSW online learning platform, your Teacher/Assessor will give you feedback via the platform.*

### Assessment outcome

Satisfactory

Unsatisfactory

**Assessor feedback**

Has the assessment declaration for this assessment event been signed and dated by the student?

Are you assured that the evidence presented for assessment is the student’s own work?

Was reasonable adjustment in place for this assessment event?

*If yes, ensure it is detailed on the assessment document.*

*Comments*:

### Assessor name, signature and date

### Student acknowledgement of assessment outcome

*Would you like to make any comments about this assessment?*

### Student name, signature and date