|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name** |  | | **Student Number** | |  | | |
| **Unit Code/s & Name/s** | ICTPRG425 | Use structured query language | | | | | |
| **Assessment Name** | **TEST** – SQL Practical Test 1 | | **Assessment Task No.** | | | 1 of 6 | |
| **Assessment Due Date** | Week 3 | | **Date submitted** | | |  | |
| **Assessor Name** | Manuela Perez [Manuela.Perez@tafe.qld.edu.au](mailto:Manuela.Perez@tafe.qld.edu.au) | | | | | | |
| **Student Declaration:**  *I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature.* | | | | | | | |
| **Student Signature** |  | | | **Date** | | |  |

|  |  |
| --- | --- |
| **Instructions to Student** | **Learning Support**  Additional support is available to help you achieve your learning goals. Speak to your teacher or a Learning Skills Centre team member if you feel that you may benefit from some extra support. The Institute provides extra support through the Disability Support Unit and the Learning Skills Centre.  RPL (Recognition of Prior Learning) is available for this unit. Speak to your teacher/assessor to check if you qualify for RPL.  **Conditions of Assessment**  You will need to complete the learning and undertake all assessments satisfactorily to be deemed competent. You are responsible for complying with all assessment item instructions; submission and collection requirements; undertaking assessment tasks honestly and retaining a copy of all assessment items.  You must submit assessment items by the **due date**, unless an extension has been granted by your teacher. Failure to submit assessment items by the due date will result in a “did not submit” being recorded and depending on your circumstances, you may be granted one final resubmission.  To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide.  **The Classroom as a Simulated Work Environment**  Students must be aware and take responsibility for the problems of working in a shared IT environment. Problems such as noise levels, production flow, interruptions and time variances are common to workplaces. In the simulated environment provided in the classroom these problems can take the form of:   * Other students who continually ask questions or talk aloud while thinking * Fire drills, projector not working, printers running out of paper or toner cartridge * Miscalculating how much work you can do in one day, missing classes and so on.   Some things are unavoidable and you must devise strategies to overcome them, for example, we cannot stop students from asking questions or entering at exiting the class. Other things are unpredictable (e.g. fire drills). You need to be aware and plan and organise your work allowing some extra time for unavoidable and unpredicted events.  **Assessment Criteria:**  To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate key skills/tasks/knowledge to an acceptable industry standard.  Refer to the marking criteria document for a detailed list of items.    **Number of Attempts:**  You will receive up to two (2) attempts at this assessment task. Should your 1st attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant sections / questions with you and will arrange a due date for the submission of your 2nd attempt. If your 2nd submission is unsatisfactory (U), or you fail to submit a 2nd attempt, you will receive an overall unsatisfactory result for this assessment task. Only one re-assessment attempt may be granted for each assessment task, with the exception of Apprentices or Trainees who are permitted an additional supplementary assessment. **For more information, refer to the Student Rules.** |
| **Submission details** (if relevant) | Submit your assessment to the allocated dropbox in **Connect** or to the allocated network folder.  Your teacher will provide all the details for the submission system or network.  Your assignment must be saved with your surname\_student number\_unit/cluster\_AssessmentNumber. For example:  **surname\_1234567890\_Database2\_1**  For re-submissions, an “R” must be added to the file name. For example:  **surname\_1234567890\_Database2\_1\_R**  The Marking Criteria Sheet must be signed and submitted with your work. |
| **Instructions to Assessor** | To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide. |
| **Note to Student** | An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide. |

# Assessment Instructions

|  |
| --- |
| EXAM CONDITIONS **Number of Questions:** 23  **Time allocated:** 2 hours  **Exam type:**  Closed Book  **Examination Conditions:** This is a practical computer-based examination.  **All** questions must be attempted.  **Materials to be supplied:** LMS (Connect)  **Materials to be supplied by the Student:**  Pen, pencil, eraser, paper for planning/drafting  **General Instructions:**  The assessor will instruct you to download and install a test database that will be used for the basis of the exam. (NB: This is not the same database used in the Learner Guide.)  You are required to answer each of the questions provided by copying and pasting your SQL statements below each question in this document.  **Exam rules:**  No mobile phones  No existing the classroom during the exam  No internet access  Use of computer restricted only to phpMyAdmin or similar software to formulate SQL statements, and Microsoft Word to copy/paste SQL statements/screenshots for submission  **Submission Instructions:** Use PHPMyAdmin, MySQL Workbench, Navicat or any other DBMS viewer available to you to create and run the queries.  Open a Word document and - for each query, provide a screenshot of the **query** and the **result.**  Make sure that the screenshots contain only of the query and results not the whole computer screen. You can use the Snipping Tool or crop the image.  When all the questions have been answered, you can upload the completed Word document to the corresponding Connect Dropbox folder. Failure to comply with the examination rules will result in your exam being invalidated. |

**Practical Exam 1**

**Submission Instructions:**

Use PHPMyAdmin, MySQL Workbench, Navicat or any other DBMS viewer available to you to create and run the queries.

Open a Word document and - for each query, provide a screenshot of the **query** and the **result.**

Make sure that the screenshots contain only of the query and results not the whole computer screen. You can use the Snipping Tool or crop the image.

When all the questions have been answered, you can upload the completed Word document to the corresponding Connect Dropbox folder.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_START of SAMPLE TEST\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q1. Create an SQL statement that retrieves all records from the **book** table.

SELECT \*

From Book

Q2. Create an SQL statement that retrieves only the **BookTitle**, **YearofPublication** and **MillionsSold** columns from the **book** table.

SELECT BookTitle, YearofPublication, MillionsSold

FROM Book

Q3. Build upon the SQL statement in Q2, by sorting the **BookTitle** by **YearofPublication** (ascending order).

SELECT BookTitle, YearofPublication, MillionsSold

FROM Book

ORDER BY `BookTitle`, `YearofPublication` ASC

Q4. Build upon the SQL statement in Q3, so that only books published after 1950 are retrieved.

SELECT BookTitle, YearofPublication, MillionsSold

FROM Book WHERE `YearofPublication` = 1950

ORDER BY `BookTitle`, `YearofPublication` ASC

Q5. Create an SQL statement to only retrieve the top 3 most recent authors in the list according to their birth year.

SELECT NAME, Surname, Nationality, BirthYear

From Author

Order BY Birthyear DESC

Limit by 3

Q6. Create an SQL statement that returns all boo rankings that have a ranking score of more than 7

Select \*

From Bookranking

Where RankingScore >7

Q7. Create an SQL statement that returns all authors where the author’ surname starts with the letter ‘T’.

SELECT \*

From Author

WHERE Surname LIKE ‘T%’;

Q8. Create an SQL statement that returns all plot source URL’s that contain the word ‘wikipedia’.

SELECT plotSource

From Bookplot

WHERE Surname LIKE ‘%wikipedia%’;

Q9. Create an SQL statement that returns all authors whose nationality is either British or Chinese.

SELECT \*

From Author

WHERE Nationality IN (‘British’, ‘Chinese’);

Q10. Create an SQL statement that uses the BETWEEN clause to retrieve all authors who died between 1925 and 1963.

SELECT \*

From Author

WHERE Death Year between 1925 and 1963

Q11. Create an SQL statement that uses the IN clause to retrieve details of all books which Genre is Novel, Fable or Fiction.

SELECT \*

From books

WHERE Genre IN (‘Novel’, ‘Fable’, ‘Fiction’);

Q12. Create an SQL statement that uses the REGEXP operator to query the book table and return all book title that either start with “T” or end with “e”.

SELECT \*

From book

WHERE REGEXP ‘^T | e$’;

Q13. Using only the author table, return a list of all unique nationalities (without duplicates).

SELECT DISTINCT `Nationality`

FROM `author`

Q14. Return a list of all authors who are still alive.

SELECT \* FROM `author`

WHERE `DeathYear` IS null

Q15. In which year did Cao Xueqin turn 18? Write a SQL query to show the name, surname and the calculated column to find the year that he turned 18.

Concatenate the name and the surname and call the column ‘Author’.

Call the calculated column “18th Birthday this Year”.

SELECT CONCAT (Name, “ ”, Surname) as Author, BirthYear+18 as ‘18th Birthday this year’

From Author

Q16. Create an SQL statement to retrieve the **oldest** and the **most recent** birth year (BirthYear) recorded in the database for authors and also calculate the **difference** between the two values.

This statement should only show a single row with the oldest year, most recent year and difference (between the two previous values).

Aliases for the columns are:

* first column “First Birth Year in DB”
* second column “Most Recent Birth Year in DB”
* third column “Years Difference”

SELECT

min (birthyear) as ‘first Birth year in DB’ Max (birthyear)

MAX(BirthYear) AS ‘most Recent Birth Year in DB’

Max(BirthYear) – MIN(BirthYear) As ‘Years Difference’

From Author

Where name = ‘cao’

Q17. Create an SQL statement to retrieve all authors from the author table. Concatenate the Name and surname as a single field. Change the column heading to “Author’s Name”.

SELECT CONCAT (Name, “ ”, Surname) AS ‘Author’s Name’

From Author

Q18. Create an SQL query that shows how many years ago was Charles Dickens born. Select, name, surname, year of birth and a calculated field to display ‘Years since Author’s Birth’.

SELECT Name, Surename, BirthYear(CURDATE()) – Birthyear as ‘years since Author’’s Birth’

FROM author

Where name = ‘Charles’

Q19. Create and SQL query to calculate and display at which age did the authors not alive today died. Display their name, surname birth year and death year and age at death.

SELECT Name, Surname, BirthYear, (DeathYear - BirthYear) AS AGE FROM author

Q20. Building on the last query, display the death age of the youngest British author.

FROM `author` WHERE `DeathYear` IS NOT null AND `Nationality` = 'british' ORDER BY `BirthYear` DESC LIMIT 1

Q21. Build an SQL statement that shows the total number of books sold (in the book table) and the average number of books sold.

This statement should only show a single row with the total and average values. Call the total column “Total Millions of Books Sold” and the average column “Average Number of Millions’.

The average result must display only 2 decimal places.

SELECT SUM(`MillionsSold`) AS `Total Sold`, AVG (`MillionsSold`) AS `Average Sold` FROM `book`

Q22. Using the GROUP BY statement, create an SQL query that lists all the nationalities of authors and the total number of authors from each nationality. Sort the nationalities alphabetically.

SELECT Nationality, COUNT(Nationality)

FROM author GROUP BY Nationality ASC;

Q23. Build upon the statement created in Q22, and use the HAVING clause to only show nationalities that have exactly 1 (one) author).

SELECT Nationality, COUNT(Nationality)

FROM author GROUP BY Nationality HAVING COUNT(\*) = 1 ORDER BY Nationality ASC

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_END of TEST\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_