

IFB130 Database Management

Project - Part B

Project overview

This IFB130 project gives you an opportunity to apply the concepts and skills you acquire in the unit to a 'realistic' database design scenario and reflect on the data requirements of an organisation.

The submission is divided into two parts due at different times during the semester. These will cover:

- A. Design of a database
- B. Creation and use of databases

The tasks for Part B

For Assessment 2B you will be required to:

- 1. Create a database for the fictitious book store *Oktomook*
- 2. Query the *Pulse Learning* Database

Weighting

Part B is worth 25 marks, for **25%** of the unit.

Groups

You can complete this assignment individually or in pairs. You do not have to work in the same pairs as you did for Part A.

If you choose to work in a pair, only one student should submit the assignment. Please provide the name and student number of the person you worked with in the README.txt file. No consideration will be given to students who claim they did more work in their pair than the other student because this assignment can be done individually.

Doing the assignment in undeclared groups, or groups larger than two students, will be treated as plagiarism. Pairs that work together and then split due to difficulties must not submit any of the same work, or it will be treated as plagiarism.

Due date

Week 12: Monday 21 May at 11.59 pm

Submission

You must submit 3 files in a ZIP file using the submission link in Blackboard:

- 1. SQL script (a text file with the file extension changed to sql) containing your solution to task 1
- 2. SQL script containing your solutions for tasks 2, 3, 4 and 5
- 3. README.txt containing your full name, student number (and the name and student number of your partner if working in pairs) and a list of any queries you have attempted but were not able to successfully run in Workbench.

Scripts in different file types will not be accepted.

Late submission

Assessment work submitted after the due date will be marked only with an approved extension (MOPP E/6.8.2). Assessment work submitted after the due date without an approved extension or, where an extension has been granted, after the extended due date, will not be marked and a grade of 1 or 0% will be awarded against the assessment item. The unit outline sets out the requirements surrounding extensions (including for disabilities), penalties for late submissions and appeals.

Assessment 2B tasks

For this assessment, you will:

1. Build a script that will create a database for a given relational schema (Week 7)
2. Provide the SQL commands needed to retrieve the required data using assessment 2a extended schema (Week 8 to 10)
3. Provide the commands to modify (INSERT, UPDATE & DELETE) the data using assessment 2a extended schema (Week 7)
4. Provide the commands needed to create appropriate indexes and views (Week 7 to 10)
5. Provide advice on the basic security measures that should be implemented (Week 11)

Task 1 [6 marks] Oktomook Book Store Database

A SQL script is a set of SQL commands saved as a SQL file. If you are already running MySQL, you can execute a SQL script file using the source command or you can import it in Workbench.

Write an SQL script that builds a database to match the relational model provided to you. These SQL statements in the script must be provided in the correct order.

Marks will be awarded for the following:

1. Creating the database (1 mark)
2. Successfully creating new tables (1 mark)
3. Including all attributes (1 mark)
4. Including constraints (1 mark)
5. Correctly creating Primary Keys (1 mark)
6. Correctly creating Foreign Keys (1 mark)

For task 1, you are required to create a database for the fictitious book store *Oktomook*. The database is based on the model below:

OKTOMOOK relational model

Customers (customerNumber, firstName, lastName, Address, city, State, postcode, region, email)

Books (ISBN, title, pubDate, PubID, Cost, Retail, Discount, category)

Orders (orderNumber, customerNumber, orderDate, shipDate, street, city, state, postCode, shipCost)

OrdersItems (orderNumber, itemNumber, ISBN, quantity, paidEach)

Author (authorID, firstName, lastName)

Wrote (ISBN, authorID)

Publishers (pubID, name, contact, phone)

FOREIGN KEYS

- Orders(customerNumber) is dependent on Customers(customerNumber)
- Orders(orderNumber) is dependent on OrdersItems(orderNumber)
- Wrote (authorID) is dependent on Author (authorID)
- Wrote(ISBN) is dependent on Books(ISBN)
- Books(pubID) is dependent on Publishers(pubID)

OTHER CONSTRAINTS

- The domain of customer(state) and orders(shipState) are [QLD, VIC, NSW, WA, TAS, NT, SA]
- ISBN must be a 13-digit number and may begin with a zero
- The publisher name and book title are both mandatory
- The default OrderItems(quantity) is 1
- The Book(category) are Fitness, Children, Computer, Cooking, Business, Literature

Task 2 [11 marks] using the Pulse Learning database

For task 2, we have provided you with the creation script for the Pulse Learning database. You must run this script in MySQL Workbench and use this database to extract the necessary information.

The script is based on the following schematic:

PULSE LEARNING relational model

Student (studentID, firstName, surname, email, streetNo, streetName, suburb, city, postcode, state, watchType, watchSerial#, buddyID)

PhoneNumber (phoneNumber, studentID)

Unit (unitID, unitName, unitCode, semester, year)

Enrolments (unitID, studentID)

UnitTutor (staffID, unitID)

Assignment (assignmentID, assignmentName, dueDate)

Grade (studentID, assignmentID, grade)

Tutor (staffID, firstName, surname, email)

SleepPatterns (sleepID, studentID, date, timeAsleep, timeAwake)

FOREIGN KEYS

- Enrolment (studentID) is dependent on Student (studentID)
- Enrolment (unitID) is dependent on Unit (unitID)
- Enrolment (staffID) is dependent on Tutor (staffID)
- StudentSleepPatterns (studentID) is dependent on Student (studentID).
- StudentSleepPatterns (sleepPatternID) is dependent on SleepPatterns (sleepPatternID)
- StudentAssignment (student ID) is dependent on Student (studentID)
- StudentAssignment (assignmentID) is dependent on Assignment (assignmentID)
- PhoneNumber (studentID) is dependent on Student (studentID)

OTHER CONSTRAINTS

- Student (state) domain is [QLD, SA, TAS, NSW, WA, NT or ACT]
- SmartWatch (type) domain is [FitBit, Suunto, Apple Watch or Samsung Gear]
- Students may have up to three phone numbers

Query 1 (1 mark)

Write a query to list the name (first and last), studentID and email of students who live in Everton Park or Everton Hills. Note: you can assume these are the only suburbs starting with 'Everton'.

Query 2 (1 mark)

Write a query to list students with buddies, in alphabetical order by surname.

Query 3 (2 mark)

Write a query to count how many units each tutor teaches. In your result-set, include the tutors' staffID, firstname, surname and the number of units they teach.

Query 4 (2 marks)

Write a query that will produce some statistics about assignment results. Your result-set should include the following:

- the assignment ID
- the assignment name
- the minimum grade achieved by the class for an assignment
- the average grade achieved by the class for an assignment
- the maximum grade achieved by the class for an assignment
- the number of submissions received.

Query 5 (2 marks)

Write a query that lists the full name and email address of all tutors that aren't teaching any classes next semester (Semester 2, 2018).

Query 6 (3 marks)

For students that slept less than 6 hours a night on average during April, list the grades for all assignments they completed during this time.

Task 3 [3 marks]

Insert (1 mark)

Write an INSERT command to insert a new unit into the Unit table. The unit is called 'Advanced Database Management', with the code IFB801 and it will be offered in Semester 1 next year (2019).

Delete (1 marks)

Write a DELETE command to remove all rows from the phone number table where the phone number starts with '02'.

Update (1 mark)

Write an UPDATE comment to change the address of all students with the last name 'Smith' who live at '180 Zelda Street, Linkburb' to '72 Evergreen Terrace, Springfield'.

Task 4 [3 marks]

Create Index (1 mark)

Currently the database only contains a small number of records. However, the data contained within it is expected to grow significantly in the future. Creating indexes on commonly searched columns is a way performance issues can be minimized.

Write a command to create an index on assignmentName of the assignment table.

Create view – 2 marks

Write a command to create a view to list the firstname, surname and student ID of any students that haven't enrolled in any units.

Task 5 [2 marks]

Pulse Learning have two employees, Nikki and Jake, to work with the MySQL database. Working as MySQL database administrator, provide the commands required to grant or revoke access so the following security requirements are met:

- A. User Nikki must be able to add records to the STUDENT table (0.5 Marks)
- B. User Nikki must be able to remove records from the STUDENT table (0.5 Marks)
- C. User Jake is no longer allowed to add data to the STUDENT table (0.5 Marks)
- D. User Jake is no longer allowed to delete records from the STUDENT table (0.5 Marks)

Assume usernames of employees Nikki and Jake are *nikki* and *jake* respectively.