Module Handbook

**Database Systems**

School of Built Environment, Engineering and Computing

2023 Level 5 Semester 1

(20 Credits)

CRN 14103

**Contents**

[1.1 Introduction from the Module Leader 4](#_Toc145326764)

[1.2 Communication 4](#_Toc145326765)

[1.3 Module Aim and Learning Outcomes 4](#_Toc145326766)

[1.3.1 Module Aims: 4](#_Toc145326767)

[1.3.2 Module Learning Outcomes: 4](#_Toc145326768)

[1.4 Graduate Attributes: 4](#_Toc145326769)

[1.5 Indicative Module Content and Learning Activities 5](#_Toc145326770)

[1.6 Module Learning Activities 5](#_Toc145326771)

[2 Weekly Schedule 6](#_Toc145326772)

[3 Key Resources to Support Learning 7](#_Toc145326773)

[3.1 Reading List 7](#_Toc145326774)

[4 Assessment Details 8](#_Toc145326775)

[4.1 Assignment Summary 8](#_Toc145326776)

[4.2 Assessment 1: Design and Report (30%) GROUP WORK 9](#_Toc145326777)

[4.3 Design of Database System – upload 6th November 2023 by 22.00 hrs 10](#_Toc145326778)

[4.3.1 Tasks 10](#_Toc145326779)

[4.3.2 Deliverables 10](#_Toc145326780)

[4.3.3 Group members rules 11](#_Toc145326781)

[4.4 Assessment 2: Implementation (70%) INDIVIDUAL 12](#_Toc145326782)

[4.4.1 Implementation 13](#_Toc145326783)

[4.4.2 Part One (40%) – upload 04th December 2023 13](#_Toc145326784)

[4.4.3 Part Two (30%) – upload 10th January 2023 and Demo 12th January 2024 13](#_Toc145326785)

[4.4.4 Deliverables 13](#_Toc145326786)

[4.4.5 Part Two - Preparation Guidance for the Demo (READ!) 14](#_Toc145326787)

[5 Reassessment/Deferral for Assessment 15](#_Toc145326788)

[5.1 Coursework 1 (30%) – Reassessment (Individual) 15](#_Toc145326789)

[5.2 Coursework 2 (70%) – Reassessment (Individual) 16](#_Toc145326790)

[6 Module Feedback 17](#_Toc145326791)

[7 Understanding Your Assessment Responsibilities 17](#_Toc145326792)

[8 Appendixes 18](#_Toc145326793)

[8.1 Appendix A: CASE STUDY LBU Care Foundation 18](#_Toc145326794)

[8.2 Appendix B: Marking Criteria – Assignment 1 (30 %) 19](#_Toc145326795)

[8.3 Appendix C: Marking Criteria – Assignment 2 (70 %): part 1 and part 2 22](#_Toc145326796)

[8.4 Appendix D: Design and Implementation (Individual task) 25](#_Toc145326797)

[8.5 Appendix E – The implementation SQL, PL/SQL and AB 31](#_Toc145326798)

[8.6 Appendix F: Group work guidelines (MAX 3 students) 32](#_Toc145326799)

[8.7 Appendix G: PEER MODERATION FORM 34](#_Toc145326800)

## Introduction from the Module Leader

Welcome to Database Systems! We hope you will find this module both challenging and rewarding. Begin by reading this guide carefully. Use it throughout DS to look up information and progress with your assessments.

We aim to provide you with a coherent set of learning opportunities, which will enable you to develop your skills and knowledge in database development stages, writing intermediate and advanced SQL statements; PL/SQL; triggers and developing back-end applications for organisational data management.

## Communication

|  |  |  |
| --- | --- | --- |
| **Module Tutor Name** | **Room** | **Email** |
| Sanela Lazarevski (Module leader) | Leighton 126 | [s.lazarevski@leedsbeckett.ac.uk](mailto:s.lazarevski@leedsbeckett.ac.uk) |

## Module Aim and Learning Outcomes

### Module Aims:

To enable students to apply database modelling and systems development skills. To apply

database skills and implement practical database solutions;

To appreciate the strength and need for procedural code when developing database system

and an application.

### Module Learning Outcomes:

LO1: Build advanced database system and applications to meet a set of user requirements using appropriate models and tools.

LO2: Identify appropriate constraints using procedural and nonprocedural database programming

Languages.

LO3: Identify, design, implement and test procedural, reusable code to meet the requirements of a given case study.

## Graduate Attributes:

|  |  |  |
| --- | --- | --- |
|  | **Developed** | **Assessed** |
| **Enterprise** |  |  |
| **Digital Literacy** | Students will apply aspects  of digital literacy in order  to research materials  relevant to the  assessments. | Y |
| **Global Outlook** | Development methods  utilised are globally  recognised. |  |

## Indicative Module Content and Learning Activities

Modern database applications: Client/Server architectures and languages; Database server programming; code reuse, data maintenance, data legislation – GDPR, Data Quality;

Data Integrity and Security: Declarative vs. procedural: advantages/disadvantages of SQL; PL/SQL - problems for which procedural code is required; triggers as constraints.

DDL, DML and SQL from SQL are fully covered on this module;

Database architecture and technologies, application tools and development approaches.

Development process: models, normalisation, physical design, testing and documentation.

Advanced Application Development: Access Control, CRUD development using Forms and Interactive Reports.

## Module Learning Activities

Lectures will be used to introduce the key concepts. Practical sessions will allow you to apply theory to practice using DBMS facilities and database application development tools. You will develop database application development skills using a workbook and a range of support exercises, examples and videos available via MyBeckett. The practical exercises reinforce the theory of data and databases and develop skills across the database development lifecycle.

Following the initial teaching week, we will support study using MyBeckett Collaborate and email. You can use the university’s specialist Computing facilities throughout the module, but we have made it possible for you to complete the module using work- or home-based computing facilities.

We welcome comments on the effectiveness of this module and any suggestions you may have for improvement using the module evaluation questionnaire or alternatively contact the Module Leader: [s.lazarevski@leedsbeckett.ac.uk](mailto:s.lazarevski@leedsbeckett.ac.uk)

# Weekly Schedule

|  |  |  |
| --- | --- | --- |
| **Week** | **Weekly topics** | **Tutorial’s topics and Self-study** |
| 1 | **Introduction to SDLC and Data Modelling, Revision and Assignment Discussion** | Consolidating learning from the previous week. Prepare for DB Requirements (session 2) learning activities. |
| 2 | **DB Requirements and Design:** Review the assessment case study looking at advanced data models (Extended Entity Relationship Models (EERM)) | Conceptual Data Models – Extended Entity Relationship Model (EERM). |
| 3 | **DB Design:** Logical Design and Physical design stage and considerations.  Feedback Opportunity on your Data Model – EERM. | Normalisation (self-study)  Read: materials in the Additional Reading Folder on MyBeckett |
| 4 | **DB Implementation:** Oracle APEX Environment; DDL, DML, Views and SQL recap | Workbook 1: APEX tasks on how to create tables, define declarative constraint and insert data; Basic and Intermediate SQL statements.  Read: materials in the Additional Reading Folder on MyBeckett |
| 5 | **DB Implementation:** SQL Advanced Statements (SQL e.g. Grouping, Outer Joins…) | SQL statements Practice Document (SQL Quiz Practice) and Views  Read: materials in the Additional Reading Folder on MyBeckett |
| 6 | **DB Implementation:**  PL/SQL – Triggers | PL/SQL  Read: materials in Additional Reading Folder  andShah Ch.10 & 11 |
| 7 | **DB Implementation:**  PL/SQL –Triggers  Feedback Opportunity so far… | Triggers, Procedure and Functions (self-study)  Read: materials in the Additional Reading Folder on MyBeckett andGault Ch.1, 2, 5, 6 Forms & Reports. |
| 8 | **DB Implementation:** Running and debugging the application; Creating APEX Reports. | Workbook 2: Forms and Reports Complete  Read: materials in the Additional Reading Folder on MyBeckett andGault Ch.1, 2, 5, 6 Forms & Reports. |
| 9 | **Data Quality and Ethics** | Complete DQ and DM tasks, and Practice on Application Builder to produce User Defined Messages in APEX, Map, Calendar, plug-ins, dynamic actions and more.  SELF STUDY: Data Quality and Data Maintenance  Read: Shah Ch. 12, 13 & 14 and Gault Ch. 9 & 10 |
| 10 | **Data Maintenance** | Complete tasks in this session’s folder. Read: Gault Ch. 7 & 8 |
| 11 | No Lecture | Support session in class |
| 12 | Support via email / online session | |

# Key Resources to Support Learning

We have made a range of useful resources available for you on MyBeckett. The Reading List can be accessed from the navigation bar.

## Reading List

**BOOKS/E-BOOKS**

Connolly, T. M. & Begg, C. E. (2015) Database Systems: A Practical Approach to Design, Implementation and Management [Online]. Harlow, Essex, England: Pearson Education Limited. Available from:

<http://ezproxy.leedsbeckett.ac.uk/login?url=http://www.vlebooks.com/vleweb/product/openreader?id=LeedsBeckt&isbn=9781292061849&uid=^u>.

[**[Nilesh](https://leedsbeckett.on.worldcat.org/search/detail/54767796?queryString=ti%3A%28Database%20systems%20using%20Oracle%3A%20a%20simplified%20guide%20to%20SQL%20and%20PL%2FSQL%29%20AND%20au%3A%28Shah%2C%20Nilesh%29&origPageViewName=pages%2Fadvanced-search-page&clusterResults=true&scope=LDS&databaseList=197%2C4096%2C1842%2C10968%2C2688%2C3036%2C2375%2C3313%2C2897%2C283%2C285%2C638)****[Shah](https://leedsbeckett.on.worldcat.org/search/detail/54767796?queryString=ti%3A%28Database%20systems%20using%20Oracle%3A%20a%20simplified%20guide%20to%20SQL%20and%20PL%2FSQL%29%20AND%20au%3A%28Shah%2C%20Nilesh%29&origPageViewName=pages%2Fadvanced-search-page&clusterResults=true&scope=LDS&databaseList=197%2C4096%2C1842%2C10968%2C2688%2C3036%2C2375%2C3313%2C2897%2C283%2C285%2C638)**](https://leedsbeckett.on.worldcat.org/search?queryString=au%3D%22Shah%2C%20Nilesh%22&databaseList=197%2C4096%2C1842%2C10968%2C2688%2C3036%2C2375%2C3313%2C2897%2C283%2C285%2C638&clusterResults=true&groupVariantRecords=false) [(2005) Database systems using Oracle : a simplified guide to SQL and PL/SQL](https://leedsbeckett.on.worldcat.org/search/detail/54767796?queryString=ti%3A%28Database%20systems%20using%20Oracle%3A%20a%20simplified%20guide%20to%20SQL%20and%20PL%2FSQL%29%20AND%20au%3A%28Shah%2C%20Nilesh%29&origPageViewName=pages%2Fadvanced-search-page&clusterResults=true&scope=LDS&databaseList=197%2C4096%2C1842%2C10968%2C2688%2C3036%2C2375%2C3313%2C2897%2C283%2C285%2C638)

Doug Gault (2015) Beginning Oracle Application Express 5.

Elmasri, R. & Navathe, S. (n.d.) Fundamentals of Database Systems [Online]. Boston: Pearson. Available from: <http://ezproxy.leedsbeckett.ac.uk/login?url=http://www.vlebooks.com/vleweb/product/openreader?id=LeedsBeckt&isbn=9781292097626&uid=^u>.

Hoffer, J. A., Ramesh, V. & Topi, H. (2019) Modern Database Management [Online]. Upper Saddle River: Pearson. Available from: <http://ezproxy.leedsbeckett.ac.uk/login?url=http://www.vlebooks.com/vleweb/product/openreader?id=LeedsBeckt&isbn=9781292263410&uid=^u>.

Pratt, P. J. & Last, M. Z. (2015a) A Guide to SQL [Online]. Australia: Cengage Learning. Available from:

<http://ezproxy.leedsbeckett.ac.uk/login?url=http://www.vlebooks.com/vleweb/product/openreader?id=LeedsBeckt&isbn=9781337028226&uid=^u>.

Pratt, P. J. & Last, M. Z. (2015b) A Guide to SQL. Australia: Cengage Learning.

**Useful web sites**

The Oracle Technet site offers useful and comprehensive support documentation. <http://technet.oracle.com>

Ask Tom – Oracle Forum [http://asktom.oracle.com/pls/APEX/f?p=100:1:0](http://asktom.oracle.com/pls/apex/f?p=100:1:0)

**Oracle Book Lists** <http://docs.oracle.com/cd/B10501_01/nav/docindex.htm>

Students requiring additional support or alternative arrangements must declare and provide evidence of their disability to the Disability Advice Team as early as possible: [www.leedsbeckett.ac.uk/studenthub/disability-advice](http://www.leedsbeckett.ac.uk/studenthub/disability-advice).

# Assessment Details

Database systems are significant within businesses and core to any computing related course (Date, 2008). You will be assessed on the application of the Software Development Life Cycle (SDLC) from analysis of data requirements through to implementation and testing stages. The database will be implemented using Leeds Beckett University’s Oracle APEX 5 system.

Your task is to design and develop a Relational Database Management System for your client at the Happy Holidays Booking system.

## Assignment Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Assignment and Deadline** | **Deliverables** | **Feedback** | **Assessed Learning Outcomes** |
| Assignment 1 (30%) – Group work Coursework: Database Design for the case study organisation – submit by 22:00 hrs on the 6th November 2023. | Upload completed template via MyBeckett | Results and Feedback via MyBeckett | LOs 1 |
| Assignment 1 (70%) – Individual  case study implementation solution and demo – submit **part 1** on the 4th December 2023 by 22:00 hrs and **part 2** by 22.00 hrs 10th January 2024. Demos on the 12th January 2024. | Upload completed template via MyBeckett and individual demo | LOs 2 and 3 |

## Assessment 1: Design and Report (30%) GROUP WORK

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module name and CRN** | | | **Database Systems** | | | |
| **Module Leader** | | | **Sanela Lazarevski** | | | |
| **Term** | 1 | **Level** | | 5 | **Approx No of Students** |  |

**ASSIGNMENT TITLE: Report and Design for Happy Holidays booking system**

**ASSIGNMENT WEIGHTING: 30%**

**HAND-OUT DATE:** September 2023

**SUGGESTED STUDENT EFFORT:** 40 hours

**SUBMISSION DATES: 6th November 2023 by 22.00 hrs**

**SUBMISSION INSTRUCTIONS:**

**FEEDBACK MECHANISM:** Individual feedback within 3 weeks via MyBeckett

**NOTE: The usual University penalties apply for late submission – please see your course handbook.**

**LEARNING OUTCOMES ADDRESSED BY THIS ASSIGNMENT:**

LO1: Build advanced database system and applications to meet a set of user requirements using appropriate models and tools.

LO3: Identify, design, implement and test procedural, reusable code to meet the requirements of a given case study**. (part of this objective)**

**NOTES:**

This is an individual assessment component. Submission of an assessment indicates that you, as a student, have completed the assessment yourself and the work of others has been fully acknowledged and referenced where needed. By submitting this assessed work, you are declaring that you are fit to submit, and you will therefore not normally be eligible to submit a request for mitigation for this work.

If your result is recorded as Non-Submission or your mark for this assessment and for the whole module is below 40%, you will have opportunity to take reassessment on the 27th March 2024 (see Reassessment information below). The mark will be capped at 40%.

If you are granted deferral through the mitigation process, you may take the reassessment demonstration with a full range of marks available. For further information, please refer to your Course Handbook or University Assessment Regulations.

## Design of Database System – upload 6th November 2023 by 22.00 hrs

### Tasks

See **Appendix A** for the Case Study details. You are expected to write a report and reflect upon all three design stages. Suggestion is that the discussion and reflection is part of each task, around 500 words.

**Task 1: Conceptual Design**

Produce an Extended Entity Relationship Model (EERM – Conceptual Design stage) for the Case Study using QSEE or SEMANTIC Pad. Discuss and argue the use of this approach for database modelling. Also, discuss and evaluate your process of data modelling using academic literature to support your discussion.

In addition, each group member complete required **tutorial tasks** and include in their own submission. The task required are found in Appendix D – Task 1.

**Task 2: Logical Design**

Produce a Relational Data Model (Logical Design stage) for the Case Study, defining all the tables including their attributes with **primary** and *foreign* keys clearly indicated using notation taught in class (see Summary of Relational Data Model rules document), based on Conceptual Design output for tasks completed for the Conceptual Design task. Discuss this stage relevance in database development cycle academic literature to support your discussion.

In addition, each group member complete required *tutorial tasks* and include in their own submission. The tasks required are found in Appendix D – Tasks 2 (A-D).

**Task 3: Physical Design**

Consider any decomposition or composition of your tables case study produced at the Logical Design stage. List any changes that you have made to the final physical model. Use literature to support your changes made and the significance of this stage in the SDLC of relational databases.

In addition, each group member complete required **tutorial tasks** and include in their own submission. The tasks required are found in Appendix D – Task 3.

**Task 4: Tables implementation**

Implement assignment case study tables (in Oracle Apex), with PKs and FKs, using QSEE function. Insert suitable test data, with 5-10 rows of data per table.

### Deliverables

You must use the **Assignment1\_template** provided to submit work by the deadline, for the Case Study (CS) tasks. This will be the same for all members. In addition to this upload each member of the team must upload tutorial tasks found in the Appendix D. In summary, these three files must be uploaded by each group Member:

1. **Assignment1\_Template (available from MyBeckett – Assessment files and feedback folder) – include group solution for the Case Study Tasks 1-4**
2. **Tutorial tasks – include individual completion of tasks in Appendix D, using Tutorial\_Upload\_Template\_2023**
3. **Peer assessment form** **– individual form completion**

### Group members rules

**There is a max of three students per group. You must choose own group by Monday of week 2, otherwise tutor will allocate you to a group. You can start on the task 1, in week 1 before groups are composed. Also, review Appendix F guidelines for working in groups and the peer assessment form, Appendix G.**

## Assessment 2: Implementation (70%) INDIVIDUAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module name and CRN** | | | **Database Systems** | | | |
| **Module Leader** | | | **Sanela Lazarevski** | | | |
| **Term** | 1 | **Level** | | 5 | **Approx No of Students** |  |

**ASSIGNMENT TITLE:** Implementation of the booking system for on Happy Holidays case study

**ASSIGNMENT WEIGHTING:** 70%

**HAND-OUT DATE:**  September 2022

**SUGGESTED STUDENT EFFORT:** 90 hrs

**SUBMISSION DATES and INSTRUCTIONS:** Upload via MyBeckett

**Part 1:** 04th December 2023 by 22.00 hrs.

**Part 2:** 10th January 2024 by 22.00 hrs with demos on the 12th January 2024

**FEEDBACK MECHANISM:** Individual feedback within 3 weeks via MyBeckett

**LEARNING OUTCOMES ADDRESSED BY THIS ASSIGNMENT:**

LO2: Identify appropriate constraints using procedural and nonprocedural database programming

Languages.

LO3: Identify, design, implement and test procedural, reusable code to meet the requirements of a given case study.

**NOTES:**

This is an individual assessment component. Submission of an assessment indicates that you, as a student, have completed the assessment yourself and the work of others has been fully acknowledged and referenced where needed. By submitting this assessed work, you are declaring that you are fit to submit, and you will therefore not normally be eligible to submit a request for mitigation for this work.

If your result is recorded as Non-Submission or your mark for this assessment and for the whole module is below 40%, you will have opportunity to take reassessment on 27th March 2024 (see Reassessment information below). The mark will be capped at 40%.

If you are granted deferral through the mitigation process, you may take the reassessment demonstration with a full range of marks available. For further information, please refer to your Course Handbook or University Assessment Regulations.

### Implementation

Based on your Assignment part 1 solution and module feedback solution (provided as part of the feedback), select two-three related tables to complete the following tasks:

### Part One (40%) – upload 04th December 2023

1. Complete required tutorial tasks and include in this submission. The tasks required are found in Appendix E. **[10 Marks]**
2. Using the tables created in Assignment 1 select 4-6 related tables for this assignment. Review these carefully and consider all types of declarative constraints for those tables and data. Create a **view** based on your table(s) and write a code to test it. Use **SQL Scripts Environment** for the whole task.

**[10 Marks]**

1. Produce a Database Application, Reports, to include **four Apex** **Reports**. Your SQL statements used to build Apex Reports should be of an intermediate and advanced level. **[10 Marks]**
2. Plan and produce **two** **Triggers** as a procedural constraint. Make sure you evidence the testing of this trigger. [**10 Marks]**

### Part Two (30%) – upload 10th January 2023 and Demo 12th January 2024

For this partyou will first need to decide which of the key database areas will be the emphasis of your database solution. Select one of the options below:

* The security of your data by implementing server and client-side solutions (e.g., User log in, password hashing, audit of user tables and data).
* Importance of programming, validations and other Apex features to ensure data quality.
* Data maintenance (batch updates) of an existing DB system.

1. Build an Application in Application Builder that will include a home page and at least one Master/Detail page that will allow CRUD functionality. Higher marks will be awarded for applications that effectively utilise a good range of Application Builder functionality depending on the database key aspects that you’ve selected to implement. Check the marking schema for guidance. [**20 Marks]**

IMPORTANT - Include clear evidence of screenshots showing your own university APEX account and student id for the Oracle APEX implementation.

### Deliverables

Submit a single compressed file, i.e., a ZIP file - **avoid a RAR file!** Any compressed file apart of ZIP will not be marked.

Evidence your work *using the Assessment2 TemplatePart1 and/or 2 (found on MyBeckett)* that is a document file with screen shots, evidencing your **student id** and where the code for the task is working successfully. (Where appropriate show evidence of code/application being tested. e.g., View images). In the zip include all file relevant to the submission. These could be all or some of the following depending on the part:

Apex Application (export application),

SQL for tables (DDL and DML),

PL/SQL (triggers),

and a copy of the actual SQL code (all apart from Application Builder code).

Part 2 is assessed **via a product demonstration** during a scheduled time in w/c 08th January 2024.

**If you do not attend the scheduled demonstration for the assessment, you will be given one further opportunity at a time which will be announced via the MyBeckett.  A 5% penalty will be applied to your mark for late demo. *If you miss the late demonstration opportunity, your mark for the assignment will be recorded as non-Submission and you will fail the module.* Note that if you upload your work but fail to attend demonstration session a mark of 0 will be recorded.**

### Part Two - Preparation Guidance for the Demo (READ!)

There will normally done in the lab (or online) one student at the time. In some cases, two tutors will mark your work for moderation purposes.

* We suggest you demo your tasks in the order of the assessment tasks.
* These are some sample questions your tutor may ask you during your demo session (if applicable):
  + Explain your PL/SQL code.
  + Demo that you can insert, update and delete data from your master/detail form.
  + Why did you choose this Chart? How is it useful? What advanced features has your report got?
  + What advanced Apex features have you implemented?
* You are being assessed on your understanding of your developed application and the underlying principles, **NOT just your ability to navigate around it**. Marks may be reduced due to poor understanding or explanation. Read marking schema.
* A demo schedule will be produced, and you should be ready to do the demo when the tutor approaches you.
* It is your responsibility to demonstrate your application in a way that provides evidence to your tutor that you have met the specification of the application, and the marking criteria provided and have a good understanding of the subject.
* **You are strongly advised to PRACTICE your demo, so that you make full use of the limited time.**
* We realise that a maximum of 7 minutes may not seem much – however, experience suggests that an application can be adequately demonstrated at this time if the demonstration is WELL PLANNED.

# Reassessment/Deferral for Assessment

Enhance the original assessment report and oral presentation (only if necessary).

Please note that each failed component (Coursework 1 (30%) and/or Coursework 2 (70%)) will have to be reassessed individually, if you have achieved a mark below 40% on the module.

## Coursework 1 (30%) – Reassessment (Individual)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module name and CRN** | | | **Database Systems** | | | |
| **Module Leader** | | | **Sanela Lazarevski** | | | |
| **Term** | 1 | **Level** | | 5 | **Approx No of Students** |  |

**ASSIGNMENT TITLE: Design**

**ASSIGNMENT WEIGHTING: 30%**

**HAND-OUT DATE:** September 2023

**SUGGESTED STUDENT EFFORT:** 40 hours

**SUBMISSION DATE:** 27th March 2024 by 22:00 hrs

**SUBMISSION INSTRUCTIONS:**

**FEEDBACK MECHANISM:** Individual feedback within 3 weeks via MyBeckett

**LEARNING OUTCOMES ADDRESSED BY THIS ASSIGNMENT:** LO1

Enhance the original assessment. An oral presentation will be organised only if necessary. As part of your submission, you are expected to submit **a summary sheet listing all changes** you’ve made in the original piece of work, if done previously; this can be easily tracked using the Word, ‘tracking’ facility.

**NOTES:**

**This is an individual assessment. Submission of an assessment indicates that you, as a student, have completed the assessment yourself and the work of others has been fully acknowledged and referenced.**

**If you miss this opportunity, your result will be recorded as Non-Submission.**

**For further information, please refer to your Course Handbook or University Assessment Regulations.**

## Coursework 2 (70%) – Reassessment (Individual)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module name and CRN** | | | **Database Systems** | | | |
| **Module Leader** | | | **Sanela Lazarevski** | | | |
| **Term** | 1 | **Level** | | 5 | **Approx No of Students** | 21 |

**ASSIGNMENT TITLE: Implementation**

**ASSIGNMENT WEIGHTING: 70%**

**HAND-OUT DATE:** September 2023

**SUGGESTED STUDENT EFFORT:** 90 hours

**SUBMISSION DATE:** 27th March 2024 by 22:00 hrs

**SUBMISSION INSTRUCTIONS:** Upload via MyBeckett

**FEEDBACK MECHANISM:** Individual feedback within 3 weeks via MyBeckett

**LEARNING OUTCOMES ADDRESSED BY THIS ASSIGNMENT:** LO2 and LO3

**NOTES:**

Enhance the original assessment – Part 1 and/or Part 2. An oral presentation will be organised only if necessary. As part of your submission, you are expected to submit **a summary sheet listing all changes** you’ve made in the original piece of work, if done previously; this can be easily tracked using the Word, ‘tracking’ facility.

**This is an individual assessment. Submission of an assessment indicates that you, as a student, have completed the assessment yourself and the work of others has been fully acknowledged and referenced.**

**If you miss this opportunity, your result will be recorded as Non-Submission.**

# Module Feedback

You receive formative feedback every week via online class. Summative assessment feedback is within 3 weeks (not including breaks/holidays) via MyBeckett.

**A mid-module review** gives you the opportunity to resolve issues with us half-way through the module. In addition, you will have the opportunity to feedback formally at the end of your module. These comments will be reviewed by your course team and may be considered at your annual course enhancement meeting. Your Course Representative will attend this and take your views to the meeting for discussion.

# Understanding Your Assessment Responsibilities

**Mitigation and Extenuating Circumstances**

If you are experiencing problems which are adversely affecting your ability to study (called 'extenuating circumstances'), then you can apply for mitigation. You can find full details of how to apply for mitigation at: [www.leedsbeckett.ac.uk/studenthub/mitigation.htm](http://www.leedsbeckett.ac.uk/studenthub/mitigation.htm).

The University operates a fit to sit/fit to submit approach to extenuating circumstances which means students who take their assessment are declaring themselves fit to do so.

**Late Submission**

Without any form of extenuating circumstances, standard penalties apply for late submission of assessed work. These range from 5% to 100% of the possible total mark, depending on the number of days late. Full details of the penalties for late submission of course work are available at [www.leedsbeckett.ac.uk/public-information/academic-regulations](http://www.leedsbeckett.ac.uk/public-information/academic-regulations).

**Academic Integrity**

Academic misconduct occurs when you yourself have not done the work that you submit. It may include cheating, plagiarism, self-plagiarism, collusion and other forms of unfair practice. What is and what is not permitted is clearly explained in *The Little Book of Academic Integrity* which is available to view at: [www.leedsbeckett.ac.uk/studenthub/academic-integrity](http://www.leedsbeckett.ac.uk/studenthub/academic-integrity).

The serious consequences of plagiarism and other types of unfair practice are detailed in section 2.9 of the Regulations at [www.leedsbeckett.ac.uk/public-information/academic-regulations](http://www.leedsbeckett.ac.uk/public-information/academic-regulations)

# Appendixes

## Appendix A: CASE STUDY LBU Care Foundation

The LBU Care Foundation is a hospital company that specialises in the care of elderly patients. There are currently four “care centres”, or hospitals, although the company does have plans to expand in this growing market of providing care to the elderly.

LBU care homes wish to invest in the development of an information system to ensure the organisation’s plans for expansion can be met and that the care centres can be effectively managed, and the quality of care assured.

Your task is to design a database that will support these requirements, taking into account the details provided below.

**Care Centres and patient treatment**

Each of the care centres is managed fairly independently, with a Medical Director taking overall responsibility for its running. The care centres deal with medical cases and employ qualified doctors and nurses to treat patients, who are generally referred by their own general practitioner, either as in-patients or out-patients.

The LBU care provides wards containing a mixture of single and twin-bedded rooms. Each bed in a ward is numbered starting at number 1. Each ward has a unique name within LBU group.

A course of patient treatment can include such elements as a course of drugs, test sessions, and a variety of therapy sessions for both body and mind. Treatment appointments may involve more than one treatment activity.

**Staffing**

LBU employs doctors, nurses and administrative staff, porters and cleaners. Staff may have different holiday and pension rights, as well as porterys and cleaners, are hourly paid.

The allocation of staff to duties is an important aspect of the efficient running of the hospital. Most staff apart from administrative staff work shifts – currently earlies, lates or nights, although some flexibility may be required in the future. Each ward has a specific requirement for nurse staff cover, which depends on the number of beds occupied, and the ward type. Each care centre has its own requirements for the number of staff required on duty.

It is anticipated that the new system will facilitate the setting up of rosters (in other words allocation of employees to shifts).

## Appendix B: Marking Criteria – Assignment 1 (30 %)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name:** | | | | **Module Tutor:** | **Total Marks: /100** | | |
| **Conceptual Model** | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |
| **30 Marks** | Excellent deployment of extended modelling skills, with complete and accurate coverage of advanced entity models, cardinality, optionality, relationship naming in accordance with the requirements.  Full and accurate list of attributes, including identifier(s), occurrences, attribute and entity definitions.  Excellent literature use to support arguments, the reflection on the tasks and learning done on this stage.  No LBU referencing style used.  Learning materials tasks completed. | All accurate entities identified and cardinality. Additional requirements specified and applied.  Full list of attributes, including identifier(s), occurrences, entity and attribute and entity definitions. Validation questions meaningful. | Mostly accurate entities identified need to address requirements with correct cardinality. Additional requirements specified and applied.  Almost a full list of attributes, including identifier(s), occurrences, entity and attribute and entity definitions. Validation questions have not much relevance to potential implementation. | Partly addressed all issues on the model, with errors in relationships naming, cardinality and wrong appliance of advance entities models.  Some consideration for entity attributes and keys, and/or cardinality.  Only few specified, with no specific validation needs  Some level of reflection done on this stage. Word count much lower than expected. No literature used.  Learning materials tasks were completed partly. | Requires further work to address errors in relationships naming, cardinality and incorrect appliance of notation.  No validation questions.  Little reflection completed. | Very little or no work done with multiple errors in relationships naming, cardinality and incorrect appliance of ERD models.  No validation questions.  No reflection done on this stage. | No work done or multiple errors in relationships naming, cardinality and incorrect appliance of ERD models.  No validation questions.  No discussion or literature used.  Learning materials tasks not completed. |
| **Logical Design** | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |
| **20 Marks** | Full list of attributes completed, attribute def., PK, all FKs correctly derived from the EERD, data values defined for each attribute meaningful, and clear evidence of M: N relations data included. Final EERD includes outputs from bottom up and top-down approaches. Excellent reflection and literature use on the tasks and learning done on this stage.  No LBU referencing style used.  Learning materials tasks completed. | Excellent decisions made for the composition and decomposition of tables, with good justification. Model is ready for the implementation. | Full list of attributes and definitions completed, PK correctly defined, all FKs correctly derived with minor anomalies, data values defined for each attribute. Decisions made for the composition and decomposition of tables have a good justification. Model is almost ready for the implementation. | List of attributes incomplete, no attributes definition, PK defined, major anomalies with FKs, no occurrences presented for each attribute. Few decisions made for the composition and decomposition of tables with no justification. Model needs changes for the implementation.  Some level of reflection done on this stage. Word count much lower than expected. No literature used.  Learning materials tasks were completed partly. | List of attributes poor, some keys defined, the majority of FKs incorrectly derived.  No decisions made for the composition and decomposition of tables with no justification. Model needs changes for the implementation | Few decisions made for the composition and decomposition of tables with no justification. Model needs major changes for the implementation.  No reflection done on this stage. | No attached composite model or no attempt made.  No discussion or literature used.  Learning materials tasks not completed. |
| **Physical Design** | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **20 Marks** | Outstanding understanding of the issues and balances of approaches of this stage. Exemplary approaches detailed and defined.  Excellent reflection on the tasks and learning done on this stage.  No LBU referencing style used.  Learning materials tasks completed. | Full list of tables and definitions completed, showing detailed and excellent consideration of appropriate Physical design rules.  Excellent literature use to support arguments, and reflection for this stage. | Full list of tables and definitions completed, showing detailed consideration of some Physical design rules.  Good reflection for this stage. | List of tables and definitions considering a few Physical design rules.  Some level of reflection done on this stage. Word count much lower than expected. No literature used.  Learning materials tasks were completed partly. | List of tables and definitions considering one Physical design rules. | Tables and definitions partly complete.  No reflection done on this stage. | Tables and definitions incomplete.  No discussion or literature used.  Learning materials tasks not completed. |
| **Tables and Data** | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |
| **30 Marks** | In addition to previous:  Fully implemented tables with PKs and FKs. The excellent population of the database with data to test the accuracy of SQL. | In addition to previous:  Data designed to show queries work effectively. | All tables implemented, column definitions very well planned. Some planning has gone into detail and all FKs constraints implemented correctly. | All tables implemented, with some anomalies in column definition. PK constraint only implemented. | Few tables implemented, with very basic data, not designed with queries in mind. Only PKs addressed, issues with FKs. | Few tables implemented, without any data and/or constraints.  Multiple errors on the FE script.  Script not running. | Not Done or a table implemented with no data population and no constraints.  *Missing clear linkage to the case study and Apex databases*. |

## Appendix C: Marking Criteria – Assignment 2 (70 %): part 1 and part 2

**Marking Schema Part 1 (40%)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name:** | | | | **Tutor marked:** | **Moderator:** | **Final Mark: /40** | | |
| **Task 1** | **Marks** | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |
| **Learning material Tasks**  **(10 marks)** | 10 | Full list of the required exercises implemented correctly with excellent discussion on each. | Full list of the required exercises implemented correctly. Very good discussion on each. | Full list of the required exercises implemented correctly with the occasional error. Very good discussion on each. | Most of the required exercises implemented or some with the small errors. Adequate discussion on each. | Many of the required exercises implemented some may have errors. Some discussion on each. | Limited list of the required exercises implemented or most with error. Poor discussion on each. | Scant number of required exercises implemented. With poor or no discussion on each. |
| **Task 2** | **Marks** | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |
| **Tables, View and Data**  **(10 Marks)** | 10 | In addition to previous:  Fully implemented all declarative constraints. View implemented and tested correctly. | In addition to previous:  Additional constraints considered: unique or check. Data designed to show queries work effectively. View implemented and testing. | In addition to previous:  Some planning has gone into detail. **View** implemented based on two tables, tested. | In addition to previous:  Some anomalies in column definition. Implemented a **view** based on one table. View not tested. | Few tables implemented, only PKs addressed, issues with FKs. Implemented a view identical to a table or view not correctly implemented. | A view not implemented.  Multiple errors on the FE script.  Script not running. | Not Done or a table implemented with no data population and no constraints.  *Missing clear linkage to the case study and Apex databases*. |
| **Task 3** |  | **80+** | **70-79** | **60-69** | **50-59** | **40-49** | **30-39** | **0-29** |
| **Apex Reports/**  **SQL**  Report 1  Report 2  Report 3  Report 4  **(10 Marks)** | 10 | Application Builder Reports with SQL Statement of an advanced level.  Apex form runs faultless. Originality in code.  Code clear with evidence of testing.  Business rule defined. | In addition to previous:  Application Builder Report’s SQL Statement based on the combination of intermediate and advanced level.  The query is based on intermediate level using build in functions (TO\_CHAR, TO\_DATE), join of 2-4 table, calculation included, necessary with sub query IN/ALL/EXSIST, GROUP BY. | Report’s SQL statement 2-3 joins. SQL Statement of an intermediate level functionality used. Query in the Apex Report is not necessary sub-query, use of DISTINCT, ORDER BY, functions (eg. MIN, MAX, AVG, COUNT).  No Application Builder done. | Apex Reports SQL statement is basic (1-2 joins and one condition in WHERE statement) and produces some inaccurate or incomplete results. | **The student used Query Builder to write SQL statements**.  Basic SQL, such as SELECT \* FROM table. | Multiple errors in the code.  No data retrieved.  Evidence of the tasks not clear. | *Missing clear linkage to the case study and Apex databases*.  Task not done. |
| **Task 4**  Procedural Constraints – Triggers  **(10 Marks)** | 10 | Code implemented are **original**, meaningful and work correctly. The code has been integrated in the Apex Application (optional).  Excellent understanding of the work presented. | There is a partial originality to the code, however code works fully. The student knows how to test package.  Excellent understanding of the work presented. | The code is covered in class; however, code works fully and correctly tested.  Good understanding of the work presented. | All code implemented runs successfully, but student is not sure how to test the code.  Moderate understanding of the work done by the student. | All code implemented runs with compilation errors.  Code cannot be tested on the data. | Attempt made, but not completed.  Code is not relevant to the case study.  The Student shows little or no understanding of the work. | Not done |

**Marking Schema Part 2 (30%) – This part is assessed via Demo**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name:** | | | | | **Tutor:** | **Moderator:** | **Final Mark: /30** | |
| **Task 5** |  |  |  |  |  |  |  |  |
| **Application Builder Pages** | 20 | In addition to previous:  There is a clear evidence that the student has **gone beyond functions and developments, that has been taught in the class and module materials**.  Features such as, LOV at the application level, Libraries, find icons, CSS, controlling access pages, connection to Website - PHP etc, publishing form… | In addition to previous: Validation fully implemented for the majority of the fields. Ajax/JavaScript used.  All features presented specified in previous sections. Messages. Plug-ins. Dynamic Actions. Pages planned well. Images integrated on the form. Security Log in features. | In addition to previous:  More than one Form implemented. Forms’ Functionality will be complete. You should consider HCI design issues (navigation-ref. data, colour, fonts, positioning, and multi-record block). Some of these features presented: item validation, default value, dynamic LOV, Map, Dash Boards. | Master-detail page. Full data manipulation available. Considered HCI design issues. Some of these features presented: Radio/check buttons, chart, default value or static LOV. Calendar. Constraints and Validation will be minimal.  . | The form is based on one table using some of the Apex features and data can be added. OR the form is master/detail and no new records. | Very limited implementation done, form not working, no data displayed or can be added. No awareness of integrity issues. Requirement not implemented. | One single form implemented and it is not running. The Student shows little or no understanding of the work. |
| **DEMO of Part 1 and Part 2** | 10 | Excellent demo. | Excellent understanding of running form will be faultless. | Understanding of running form will be faultless. | Demonstration of the requirement will be competent. | The student shows some understanding. | Student shows no understanding of the code demoed to tutor student is not able to answer correctly any of the questions | DEMO not done. |
| **General Feedback:** | | | | | | | | |

## Appendix D: Design and Implementation (Individual task)

Complete all tasks and evidence work as suggested in the Tutorial\_Tasks upload document.

**Task 1 – Extended Entity-Relationship Model**

**Holiday Package**

Each holiday package is provided for a certain number of customers. Each holiday package is categorised as being of a specific type e.g. Beach, Winter Sun, Skiing and Adventure, each of which have specific, unique requirements. Each package will fall into one of two journey types, either

* Return flight to a single destination
* Return coach transport with two destinations.

Holidays may have a courier, although some do not.

Where group bookings are made the company needs to record information about the member of the group who is the Group leader; i.e. the arranger of the holiday.

Step 1: Underline all nouns in the text above and list them below:

Step 2: Ask yourself - Is your noun

* an attribute of an entity, or
* an occurrence of an attribute or
* an entity, which would have more than ONE occurrences and is it relevant to our system (case study)?

Step 3: For each entity create a table to defining attributes and occurrences.

Example: Holiday Package

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute’s Name | Number | Customer | Type | Journey Type |  |  |  |  |
| Occurrences | Beach | Winter Sun | Skiing | Adventure |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |

Example: Type

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute’s Name | Requirement |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |

Example: Journey

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute’s Name | Return flight | Single destination | Coach transport | Two destinations |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |

Example: Group Bookings

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute’s Name | Information |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |

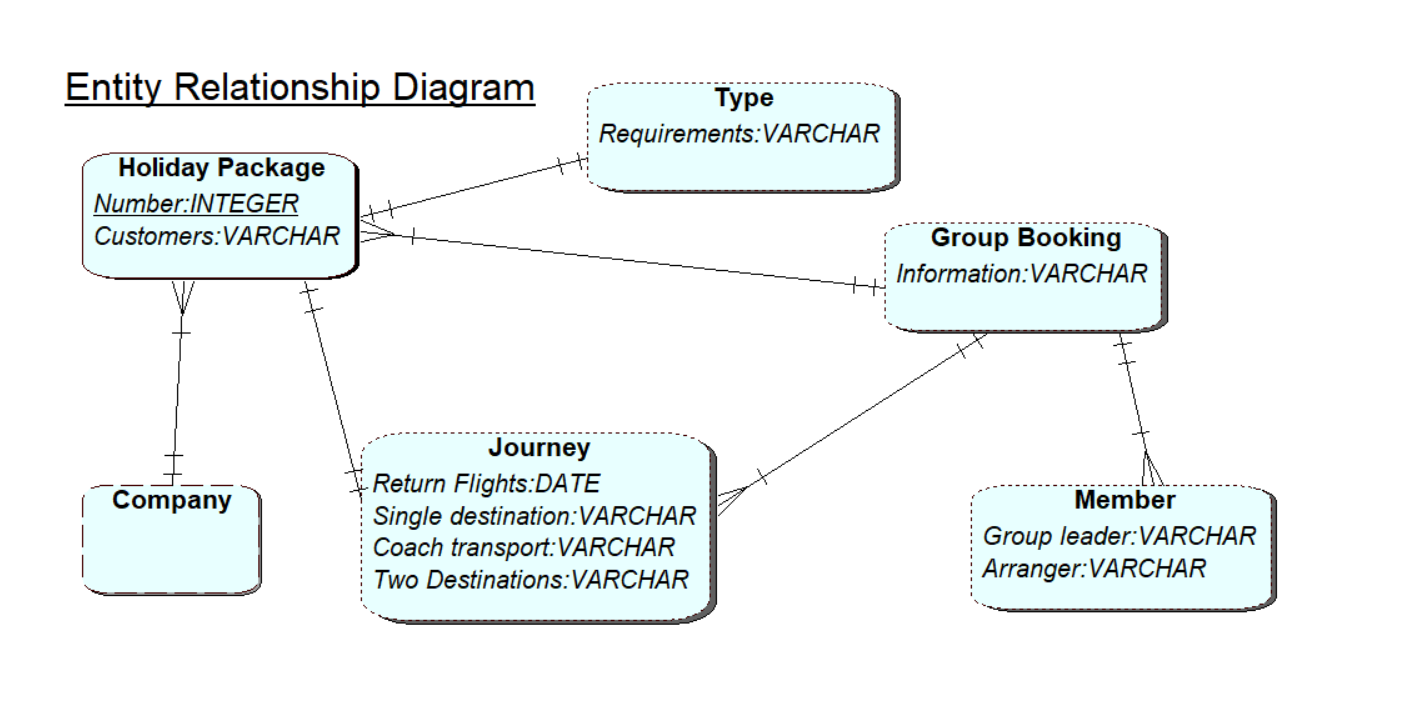
Example: Company

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute’s Name |  |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |

Example: Member

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute’s Name | Group leader | Arranger |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |
| Occurrences |  |  |  |  |  |  |  |  |

Step 4: Draw your ERD, by defining entities, relationships, relationship names cardinality.



Task 2 (got A, B, C activities) - Logical Design and Normalisation

**A.**

Fill in the missing Entity names.

**Customer**(Custid,

**Order** (Orderid, ….*.Custid*



Customer Order

**Order** (Orderid,

**Item** (Itemid,…*Order\_id*



Order Item

**Order** (Orderid,

**Product** (Product\_id

**Item** (Itemid,…***Order\_id, Product\_id***



**Order Item Product**

1. In this task you are asked to produce an ERD using the given relations and the keys (logical design in reverse).

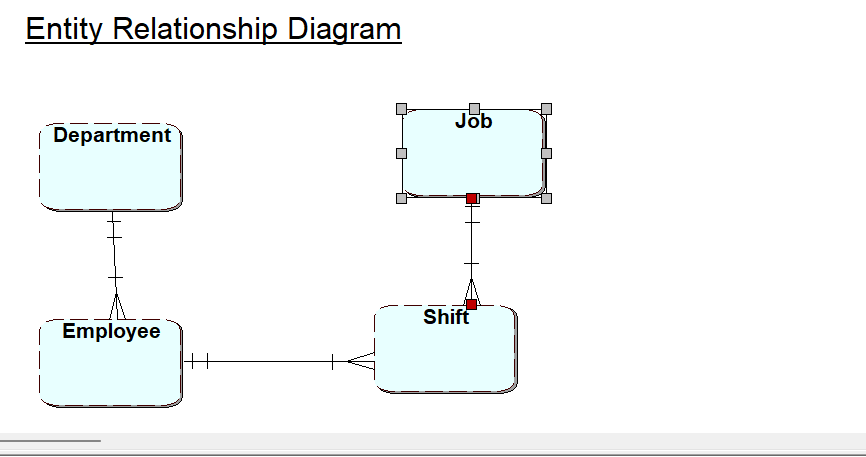
**Department** (Dept\_id, …)

**Employee** (Emp\_id,….., *Dept\_id)*

**Job** (Job\_id,….)

**Shift (***Job\_id, Emp\_id,….)*

****

****

**B.**

Derive this model fully by defining ***foreign keys*** for the relevant tables.

Below is the ERD’s description:

*Every project requires a number of employees. It is usual for consultants to work on more than one project at a time. Each project is broken down into individual tasks which are allocated to specific consultants. Some tasks are common to any project. Consultants meet regularly with clients by making appointments with them.*

Note: When you are completing the task to derive keys from your ERD, no retrospective changes should happen to the ERD design.

Diagram

Description automatically generated

**Derived Tables following Relational Data Model(RDM) rules**

**Step 1: Identify Relations/tables and attributes**

**Step 2: Identify Primary key(s) for each relation/table**

**Step 3: Derive Foreign Keys following the RDM rules**

**Steps 1 and 2 have already been completed for you below.**

Step 3:

Task(Task\_id, Task\_Name,…

Project( Project\_id, Project\_StartDate, Project\_EndDate,…

Project\_Task(PT\_id, Desc,…

Consultant(Consultant\_id, Name, Address, Phone,…

Client (Client\_id, Name, Address, Phone,…

Appointment(Appointment\_ref, Date, Location,…

**C. HOUSE LETS – Normalisation**

The Agency undertakes regular inspections of the properties. Staff are allocated a company pool car for the day. A member of staff may inspect many properties in one day, but a property is inspected only once in any one day.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Page No: 1 | | Acme Letting Agency  Property Inspection Report | | | | | Date: | 12/10/96 | |
| Property No: p103 | | | | Property Address: | | | | | |
| Inspection Date | Inspection Time | | Comments | | Staff Number | Staff Name | | | Car Reg. |
|  |  | |  | |  |  | | |  |
|  |  | |  | |  |  | | |  |

Your task is to identify which Normal Form has been incorrectly normalised.

|  |  |  |  |
| --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** |
| Property No. | Property No. | Property No. | Property No. |
| Property Addr. | Property Addr. | Property Addr. | Property Addr. |
| Idate\* |  |  |  |
| Itime \* | *Property No.* | *Property No.* | *Property No.* |
| Comment\* | Idate | Idate | Idate |
| Staff\_no\* | Itime | Itime | Itime |
| Sname\* | Comment | Comment | Comment |
| Car\_reg\* | Staff\_no | *Staff\_no* | *Staff\_no* |
|  | Sname |  |  |
|  | Car\_reg |  |  |
|  |  | Staff\_no | Staff\_no |
|  |  | Sname | Sname |
|  |  | Car\_reg | Car\_reg |

**Task 3** **- Physical Design**

**Complete the Physical Desing SQL tutorial – based on scott tables.**



1. Notice departments are in different cities. (horizontal partitioning)
   1. Write SQL to create 4 ‘emp’ tables, one for each city eg tmpempny, tmpempda. Check you haven’t lost any data. Create this as a script (drop tables before creating to make it re-runnable).
   2. 1a) Now write SQL to select all employees from the Company.
   3. Hint: use the UNION command. UNION compatibility is important.
2. Assume we need to hold sensitive employee data on one table ‘tmpempprsnnl’ and non sensitive data on another table tmpempcommon.

Write the SQL to do this. Check the data is correct. Create it as a re-runnable script. (vertical partitioning).

1. 2a) Now write SQL to select all details of all employees from tmpempprsnnl and tmpempcommon.
2. Do a combination of the above. There should be one tmpempprsnnl table containing all the sensitive data for all the employees and 4 other emp tables containing the general emp data (tmpempny, tmpempda etc). This is combined partitioning.

**Include clear evidence of screen shots showing your own APEX account and student id for the Oracle APEX implementation.**

## Appendix E – The implementation SQL, PL/SQL and AB

1. Workbook Tasks

Compete the following Exercises from the SQL Workbook and include a discussion of the SQL code including how it works and arrives at the output given:

1. Exercise 14 Page 19
2. Exercise 18 Page 19
3. Exercise 29 Page 24/25
4. Exercise 35 Page 27
5. Exercise 36 Page 28
6. Exercise 40 Page 29

2. PLSQL - Trigger Tasks

Complete Exercises 3 and 4 from the PLSQL – TRIGGERS Workbook. Ensure you test it appropriately and provide a discussion on value of each trigger and how the testing you have used ensures a full testing strategy.

3. Application Builder Task

Complete the following Exercises from the Application Builder Workbook and include a discussion of what you have created and how it works.

1. Create an Application that includes the following:
2. a Master Detail form - you may choose the Master Detail type with an LOV and Calendar.
3. an Interactive report - demonstrate form of at least two actions.
4. two Charts with a clear title.

## Appendix F: Group work guidelines (MAX 3 students)

Group members will be those in your tutorial group.

**Have respect for each other.**  
- Respect each other’s ideas  
- Respect the other group members  
- Don’t interrupt each other  
- Everyone’s opinion should count  
- Be honest with each other

**All group members should do an equal amount of work.**  
- Everyone should share the responsibility of the tasks  
- Don’t take over and don’t let others take over

Your group should have a common understanding of goals that need to be achieved.  
- Help each other to understand all concepts

**Be open to compromise.**  
- Be willing to cooperate with others on their ideas  
- Keep an open mind  
- Vote on disagreements

**Effective communication.**  
- Make sure everyone is able to be vocal about their ideas and problems  
- Give ideas no matter how “off” you may think they are  
- Listen effectively  
- Don’t be critical

**Time management.**  
- Attend and arrive on time to all group meetings  
- Be flexible about meeting times  
- Keep on task (limit talk about non-related events)

Be happy in the group you are in.

**Nonparticipating group members process**

The following documents the procedures for dealing with non-participating members

We anticipate that in most cases all group members will participate fully in the assignment and that there will be no problems. However, in order that all eventualities are catered for this procedure exists to deal with any cases where members of a group are not contributing to the progress of the group.

Since you are unable to “sack” members from your group, you must adhere to the following procedures to ensure that if any such cases arise, the group as a whole is not disadvantaged;

If a group member’s lack of engagement in the assignment is causing a problem the group should initially attempt to solve the problem inside the group.

If these efforts are unsuccessful the group should see their tutor with evidence of the non-involvement (for example attendance lists for meetings or non-delivery of agreed work).

If this evidence is sufficient the tutor will send a formal letter to the requiring the student to attend a one-to-one meeting with the tutor.

At this meeting the reasons for the student’s non-performance will be ascertained and the actions he or she needs to take to be reinstated in the group will be determined.

A student who does not attend this meeting without good reason, or who, after this meeting, still fails to make a reasonable contribution will have to accept the allocation of marks given by the peer assessment process, by completing the peer assessment form at the point of submission by all members of the team.

## Appendix G: A picture containing text, graphics, graphic design, logo Description automatically generatedPEER MODERATION FORM

For this assignment you have worked in a team. You now have the   
opportunity to score each team member's contribution to the project.

Why Peer Moderation?

This is part of your own professional development as it helps you to be critical of others' work and understand how to work together cooperatively. Peer Moderation may affect individual marks; it thereby ensures that every team member is awarded a fair grade for their work.

How to do it

If all team members have produced an equal amount of work then you would score everyone Zero. This score would not affect the mark of any member of the group.

If however, one person has done less work, you may score them at '-5' (for instance). The total score for the whole group must equal ZERO though. If you give one member of the group a negative score, you have to score the other group members positively until the sum of all scores equals ZERO.

Shall I discuss this within my group?

No. Peer Moderation is confidential – do not share your scores with your group members, and do not ask them how they have scored you or others.

In 7 easy steps…

1. Put your name in the box below
2. Put your group members' names in the first column of the table
3. Score all your group members - Do not rate yourself!
4. Use scores in the range from -10 to +10
5. All scores have to add up to ZERO
6. If a team member has made no contribution then score them NC.
7. Submit on the date the portfolio is due for submission

**YOUR NAME:**

|  |  |  |
| --- | --- | --- |
| **THE OTHER GROUP MEMBERS** | | **SCORE (-10…10)** |
|  | |  |
|  | |  |
|  | |  |
| **TOTAL** | **0 (ZERO)** |