**Coursework Details**

**The coursework assignment is an individual assessment weighted 40% of the marks for the module. It is designed mainly to assess students’ practical problem-solving skills and critical thinking/evaluation on the design and development of database systems. It requires the student to analyse, design, and implement a web-based database application based on a given business case study. You are asked to provide a software solution as well as appropriate documentation detailing the design and implementation of the system.**

Sure, I can help you with that. What specifically do you need help with?

1. **Case Study**

**An organization is planning to establish online voting system to choose employee of the month. Employees are not allowed to vote themselves also one employee can work on one department only but allowed to vote an employee from any department. There is different role of employees in different department, and everyone is eligible to vote. In Organization, each department have their own manager who are also an employee of that organization and organization keep the record of employee job history if any employee changes his/her department and job. Your prototype of the system will be developed using Oracle SQL Developer Data Modeler and ASP.NET with C#.**

Thank you for providing the case study details. Here is a possible approach to completing the coursework assignment:

1. Analyze the requirements of the case study: You will need to carefully read and analyze the given case study to understand the requirements of the online voting system. Take notes of key points, constraints, and assumptions. Make sure to identify the entities, relationships, and business rules involved in the system.
2. Design the database schema: Using Oracle SQL Developer Data Modeler, design the database schema for the online voting system based on the requirements. Identify the entities, attributes, primary and foreign keys, and relationships between entities. Create ER diagram to visualize the schema.
3. Implement the database: Using SQL, create the tables, views, indexes, and other database objects based on the schema. Populate the database with sample data to test the system.
4. Design and implement the user interface: Using ASP.NET with C#, design and implement the user interface for the online voting system. This will involve creating web pages, forms, and controls to allow users to interact with the database. Ensure that the user interface is intuitive, easy to use, and meets the requirements of the case study.
5. Test and debug the system: Test the system thoroughly to ensure that it meets all the requirements and works as expected. Debug any issues or errors that arise during testing.
6. Write documentation: Write documentation that details the design and implementation of the system. This should include the database schema, SQL code, ASP.NET code, and user manual. The documentation should be clear, concise, and easy to understand.
7. Submit the coursework: Submit the completed coursework, including the documentation, to your instructor for grading.

I hope this helps you in completing the coursework assignment. Let me know if you have any further questions or need any assistance.

**Fig 1.**

**Example of Employee Details S.N. Employe e Name Date of Birth Contact Email Address Address Departmen t 1 Erling Haland 1990- Aug-10 987657876 8 Erling@gmail.com New Baneshwor, Kathmandu, Nepal Finance 2 Dejan Kulusevs ki 1988- Sep-20 983747627 4 Dejan@gmail.com Begnas Lake, Pokhara, Nepal Human Resource 3 Lisandro Martinez 1989- Jan-12 987656565 6 Lima@gmail.com Itahari, Koshi, Nepal Sinamangal, Kathmandu, Nepal Marketing 4 Raphael Varane 1990- Feb-14 981237309 0 Raphael@gmail.co m Dipileush@gmail.co m Sinamangal, Kathmandu, Nepal Human Resource**

Thank you for providing an example of employee details. This will help in understanding the structure and content of the database table. Based on this example, the following table could be created in the database:

Employee Table:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| employee\_id | integer | Unique identifier for each employee |
| employee\_name | varchar(50) | Name of the employee |
| date\_of\_birth | date | Date of birth of the employee |
| contact\_number | varchar(15) | Contact number of the employee |
| email\_address | varchar(50) | Email address of the employee |
| address | varchar(100) | Address of the employee |
| department | varchar(50) | Department to which the employee belongs |

Using the above table structure, you can create an SQL code to create the Employee table in your database. After the table is created, you can insert sample data from the given example to test the system.

**Fig 2.**

**Example of Voting Record Voter ID Voter Name Voting Year Voting Month Candidate ID Candidate Name Candidate Department 1 Erling Haland 2019 January 88 Nate Diaz IT 1 Erling Haland 2019 February 132 Kamaru Usman Finance 1 Erling Haland 2020 August 420 Dana White Human Resource 1 Erling Haland 2020 September 7 khabib nurmagomedov Finance**

Thank you for providing an example of the voting record. Based on this example, the following table could be created in the database:

Voting Table:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| voter\_id | integer | Unique identifier for each voter |
| voter\_name | varchar(50) | Name of the voter |
| voting\_year | integer | Year in which the voting took place |
| voting\_month | varchar(15) | Month in which the voting took place |
| candidate\_id | integer | Unique identifier for the candidate |
| candidate\_name | varchar(50) | Name of the candidate |
| candidate\_department | varchar(50) | Department to which the candidate belongs |

Using the above table structure, you can create an SQL code to create the Voting table in your database. After the table is created, you can insert sample data from the given example to test the system.

**2. Requirements of the Coursework Marks are awarded for producing a working and properly documented system that meets the requirements specified below as deliverables:**

Based on the information provided, the following requirements can be identified as deliverables for the coursework:

1. Analysis Report: An analysis report detailing the business case study, user requirements, and system requirements for the web-based database application. The report should also identify the main entities and relationships in the database and describe the functionalities that the system should provide.
2. Entity Relationship Diagram: An entity relationship diagram (ERD) depicting the entities, attributes, and relationships in the database. The ERD should be created using Oracle SQL Developer Data Modeler.
3. Database Implementation: A working database system that implements the requirements specified in the analysis report. The database should be implemented using Oracle SQL Developer, and the implementation should be well-documented, including SQL code to create the necessary tables and populate them with sample data.
4. Web-based User Interface: A web-based user interface that allows employees to vote for the employee of the month. The user interface should be implemented using ASP.NET with C# and should provide the following functionalities:

* Display the list of employees who are eligible to be voted for.
* Allow employees to cast a vote for an employee of their choice.
* Display the voting results.

1. User Manual: A user manual that provides instructions for using the web-based database application. The manual should include screenshots and step-by-step instructions for each functionality.
2. Technical Report: A technical report that documents the design and implementation of the database system and the web-based user interface. The report should include details about the design decisions, database schema, implementation details, and testing procedures. It should also include any challenges faced during the implementation and how they were overcome.

The above requirements should be met to produce a working and properly documented system, and the marks will be awarded based on the quality of the deliverables.

1. **Requirements of the Coursework**

**Marks are awarded for producing a working and properly documented system that meets the requirements specified below as deliverables: 2.1 Contents Page A list of sections/subsections of the document, including page numbers. 2.2 Normalisation [15 Marks] Produce a set of fully normalised tables for the system: ● You may use Figure 1 and Figure 2 as a starting point for normalisation. ● You may also add additional attributes where appropriate. ● Show clearly all the steps of normalisation, up to the 3rd normal form. ● Two separate normalisation is done showing the correct transition between UNF to 3NF. ● Proper identification of Primary/Foreign Key, Repeating Groups, Partial Dependency, and Transitive Dependency**

2.1 Contents Page The Contents page should list all the sections and subsections of the document, along with their page numbers.

2.2 Normalisation [15 Marks]

The aim of this task is to produce a set of fully normalised tables for the system. You may use Figure 1 and Figure 2 as a starting point for normalisation. You may also add additional attributes where appropriate. You should show clearly all the steps of normalisation, up to the 3rd normal form. Two separate normalisations should be done, showing the correct transition between UNF to 3NF. Proper identification of Primary/Foreign Key, Repeating Groups, Partial Dependency, and Transitive Dependency should be provided.

Normalization is the process of organizing data in a database so that it is consistent and efficient. It involves breaking down a table into smaller tables and establishing relationships between them. The process results in a set of tables that are in a specific normal form, which is determined by a set of rules. There are several normal forms, but we will focus on the first three:

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)

The following sections will detail the steps taken to normalize the data in Figure 1 and Figure 2.

2.2.1 Normalization of Employee Details The first step in normalizing the employee details table is to identify the repeating groups. From Figure 1, it can be seen that there are multiple instances of the same information for each employee.

To remove the repeating groups, we need to create a new table for each distinct type of information. This results in the following tables:

* Employee (Employee ID, Employee Name, Date of Birth, Contact, Email Address, Address)
* Department (Department ID, Department Name)
* Job History (Employee ID, Department ID, Job Start Date, Job End Date)

The Employee table is now in 1NF, as each column contains atomic values.

To achieve 2NF, we need to identify the partial dependencies. A partial dependency occurs when a non-key attribute is dependent on only a part of the primary key.

In the Job History table, the Job Start Date and Job End Date are dependent on the Employee ID and Department ID. Therefore, we can split the table into two tables:

* Employee Department (Employee ID, Department ID)
* Job History (Employee ID, Department ID, Job Start Date, Job End Date)

The Employee Department table now contains the composite primary key of Employee ID and Department ID. The Job History table has a foreign key constraint to the Employee Department table.

To achieve 3NF, we need to identify the transitive dependencies. A transitive dependency occurs when a non-key attribute is dependent on another non-key attribute.

In this case, there are no transitive dependencies. Therefore, the tables are already in 3NF.

The final normalized tables for Employee Details are:

* Employee (Employee ID, Employee Name, Date of Birth, Contact, Email Address, Address)
* Department (Department ID, Department Name)
* Employee Department (Employee ID, Department ID)
* Job History (Employee ID, Department ID, Job Start Date, Job End Date)

2.2.2 Normalization of Voting Record The first step in normalizing the voting record table is to identify the repeating groups. From Figure 2, it can be seen that there are multiple instances of the same information for each voter.

To remove the repeating groups, we need to create a new table for each distinct type of information. This results in the following tables:

* Voter (Voter ID, Voter Name)
* Voting Year (Voting

continue

2.3 Entity Relationship Diagram [20 Marks] Produce an ERD for the database system. Your ERD should: ● Show all the entities, attributes, and relationships in the system. ● Use correct notation for entities, attributes, and relationships. ● Indicate the cardinality and participation of each relationship. ● Indicate the primary and foreign keys of each entity. ● Ensure that your ERD is in 3rd normal form.

2.4 Logical Data Model [10 Marks] Produce a Logical Data Model (LDM) of the system, based on your ERD. Your LDM should

more from 2.2.2

Sure, here are some more details for 2.2.2:

To show the steps of normalization up to the 3rd normal form, you will need to follow the guidelines and rules of normalization. Here is a brief overview of each normal form:

1. First Normal Form (1NF): Each table should have a primary key, and each column should contain atomic values (i.e., indivisible values that cannot be further subdivided).
2. Second Normal Form (2NF): All non-key attributes of a table should be functionally dependent on the entire primary key, not just a part of it.
3. Third Normal Form (3NF): All non-key attributes of a table should be dependent only on the primary key, and not on any other non-key attributes.

To perform normalization, you will need to start with an unnormalized table (UNF) and apply the rules of normalization step-by-step to produce a set of normalized tables. You will need to identify repeating groups, partial dependencies, and transitive dependencies, and then eliminate them by splitting the tables or creating new ones.

Once you have produced a set of normalized tables, you will need to identify the primary keys and foreign keys for each table. The primary key is a unique identifier for each row in the table, and the foreign key is a reference to a primary key in another table.

Overall, you should aim to produce a set of normalized tables that are free from any redundancy or inconsistencies, and that represent the data in a clear and organized manner.

give ma UNF-3NF of voting table

Sure, here's an example of the normalization process for the voting table:

UNF (Unnormalized Form)

* Voter ID (PK)
* Voter Name
* Voting Year
* Voting Month
* Candidate ID
* Candidate Name
* Candidate Department

1NF (First Normal Form)

* Voter ID (PK)
* Voter Name
* Voting Year
* Voting Month
* Candidate ID
* Candidate Name
* Candidate Department

2NF (Second Normal Form)

* Voter ID (PK)
* Voter Name
* Voting Year (PK)
* Voting Month (PK)
* Candidate ID (PK)
* Candidate Department

3NF (Third Normal Form)

* Voter ID (PK)
* Voting Year (PK)
* Voting Month (PK)
* Candidate ID (PK)
* Candidate Department (FK)

In the final 3NF table, we have removed the transitive dependency between Candidate ID and Candidate Department. Now, Candidate Department is dependent only on Candidate ID and not on any other non-key attribute.

**2.3 E-R Model [10 Marks] Use Oracle SQL Developer Data Modeler to produce an Entity Relationship Diagram. The final ERD should be consistent with the outcome of your normalisation. Submit a copy of the ERD: ● Proper ERD of the textual description with proper entities and correct cardinality (entities must show all primary keys and foreign keys involved). ● Explanation of assumptions made in order to make the ERD (must show the process to remove the duplication of entities(relations) from Relational Model, Normalization 1 and Normalization 2)**

Based on the normalized tables, the following ERD is proposed:



Assumptions made in order to create the ERD:

* Each employee can vote for multiple employees.
* Each voting record is associated with a voter and a candidate.
* A candidate can receive multiple votes in different voting periods.
* A department can have multiple employees, and each employee can belong to only one department.
* An employee can have multiple job histories.
* Each job history record is associated with one employee and one department.
* Each department has one manager, and each manager is also an employee of the organization.

**2.4 Data Dictionary [5 Marks] Use Oracle SQL Developer Data Modeler to produce a list of attributes for each entity. Submit a print-out copy of these lists: ● Data Dictionary must contain well-defined Name of Tables, Attributes, Appropriate Data Type and Size of Attributes, Constraints of Each attribute, Reference Tables and Attributes along with Example Data.**

The following is an example of a data dictionary for the Employee table:

Table Name: Employee Attributes:

* Employee ID (Primary Key, integer)
* First Name (varchar2(50))
* Last Name (varchar2(50))
* Date of Birth (date)
* Contact Number (varchar2(20))
* Email Address (varchar2(100))
* Address (varchar2(200))
* Department ID (Foreign Key referencing Department table, integer)

Example data:

| **Employee ID** | **First Name** | **Last Name** | **Date of Birth** | **Contact Number** | **Email Address** | **Address** | **Department ID** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Erling | Haland | 1990-08-10 | 9876578768 | [Erling@gmail.com](mailto:Erling@gmail.com) | New Baneshwor, Kathmandu, Nepal | 1 |
| 2 | Dejan | Kulusevski | 1988-09-20 | 9837476274 | [Dejan@gmail.com](mailto:Dejan@gmail.com) | Begnas Lake, Pokhara, Nepal | 2 |
| 3 | Lisandro | Martinez | 1989-01-12 | 9876565656 | [Lima@gmail.com](mailto:Lima@gmail.com) | Itahari, Koshi, Nepal | 3 |
| 4 | Raphael | Varane | 1990-02-14 | 9812373090 | [Raphael@gmail.com](mailto:Raphael@gmail.com) | [Dipileush@gmail.com](mailto:Dipileush@gmail.com) | 2 |

Note: The Department ID attribute is a foreign key referencing the Department table, which will be included in the data dictionary for the Department table.

* 1. **Generation of Database [3 Marks+ 4 Marks+ 3 Marks] ∙**

**Use Oracle SQL Developer Data Modeler to convert the E-R diagram into a set of database tables. Provide a print-out of the DDL script for generating the tables (relevant ‘CREATE statements only). ∙ Use Oracle SQL Developer to populate these tables with suitable data values (using ‘INSERT’ statements), at least 5 rows for SETUPS, and 10 rows for CONFIGURATION and TRANSACTION tables with proper screenshot. ∙ Provide a print-out of contents for all the tables (using ‘SELECT’ statements) with a proper screenshot.**

I am sorry, but as an AI language model, I do not have access to the file or database system of your computer to provide screenshots. However, I can provide you with guidance on how to complete this part of the coursework.

To generate the database, you can use the DDL script generated by Oracle SQL Developer Data Modeler. This script should contain the 'CREATE' statements for all the tables, along with their attributes, data types, and constraints.

Once you have the DDL script, you can use Oracle SQL Developer to execute the script and create the tables in the database.

After creating the tables, you can populate them with suitable data values using 'INSERT' statements. You can use SQL Developer to execute these statements and insert the data into the tables.

To verify that the data has been inserted correctly, you can use 'SELECT' statements to retrieve the data from the tables. You can use SQL Developer to execute these statements and display the contents of the tables.

Make sure to take proper screenshots of the SQL Developer window while executing the 'INSERT' and 'SELECT' statements, as well as the resulting data in the tables. These screenshots will serve as evidence of the successful generation and population of the database.

**2.6 Implementation of Web-based Database Application ❑ Implementation of a web-based database application which includes the following webforms (web pages) using ASP.NET with C#: ● Basic Webforms: [15 Marks] ⮚ Employee Details ⮚ Department Details ⮚ Job Details ⮚ Role Details ⮚ Address Details All these forms should facilitate input, update, and delete of information. ● Complex Webforms: (Provide SQL Queries and Complex forms) [6+14 Marks] ● Employee Job History Mapping Form (for any Employee, show details of the employee and the details of all the previous department, job that he/she has done with job time period) Note: - You are not allowed to show the current working department and job ● Voting Record Form (For any Employee, show the detail of the employee and all the voting record that he/she have given including the details of candidate.) ● Employee of the Month Form (for any month, show the details of top 3 candidate who received the highest vote). ❑ Implementation of a homepage website that includes an options menu with an Attractive Graphical Dashboard.**

It seems that you would like me to provide information on the requirements for the implementation of a web-based database application. Here are the details:

2.6 Implementation of Web-based Database Application [35 Marks] The implementation of a web-based database application includes the following webforms (web pages) using ASP.NET with C#:

● Basic Webforms: [15 Marks] ⮚ Employee Details ⮚ Department Details ⮚ Job Details ⮚ Role Details ⮚ Address Details

All these forms should facilitate input, update, and delete of information.

● Complex Webforms: (Provide SQL Queries and Complex forms) [6+14 Marks] ● Employee Job History Mapping Form (for any Employee, show details of the employee and the details of all the previous department, job that he/she has done with job time period)  
Note: - You are not allowed to show the current working department and job ● Voting Record Form (For any Employee, show the detail of the employee and all the voting record that he/she have given including the details of candidate.) ● Employee of the Month Form (for any month, show the details of top 3 candidate who received the highest vote).

In addition to the webforms, a homepage website that includes an options menu with an Attractive Graphical Dashboard should be implemented.

The implementation should be done using ASP.NET with C#. It should facilitate input, update, and delete of information and incorporate SQL queries and complex forms for the employee job history mapping form, voting record form, and employee of the month form.

The implementation should be properly documented, and the documentation should include an explanation of the assumptions made during the implementation process. The implementation should also be properly tested to ensure that it is functioning as expected.

**2.7 Documentation of the system (as implemented in 3.6) FOR EACH FORM Implementation document ● Provide a set of screen dumps for all the web pages (webforms) you have produced. ● Basic Forms to show CRUD operation (form view and list view screens with Template Fields for Foreign Keys)) ● Complex Forms (Proper Filter Demonstration using List Box/Grid and Template Field (foreign keys)) Testing Document [10 Marks] ∙ Provide a copy of the initial data (table contents) in your system. ● For each form implemented, list the individual tests that have been carried out together with their results. ● Proper test cases with Before and After Screenshots of data ● At least 2 failure cases with proper correction measures FOR THE APPLICATION The URL address for the uploaded website (connected with the Oracle database) User Manual (up to 5 pages) [5 Marks] ∙ The User Manual should have a contents page and separate sections for each form provided. ∙ The User Manual should contain clear instructions on how to use the system and how to run each of the forms available to it. ● Easy to read user manual with Arrows and Graphics to explain the process**

It seems like you have provided the requirements for a coursework assignment. Is there anything specific you would like to ask me regarding these requirements?

**2.8Further Discussion [5 Marks] Your discussion should summarise your experience in undertaking this coursework with the mention of 5 tools/techniques learned during coursework.**

As an AI language model, I don't have personal experience in undertaking coursework. However, I can provide some common tools/techniques that can be used for this coursework:

1. Oracle SQL Developer Data Modeler: A tool for designing and modeling database schema.
2. SQL (Structured Query Language): A programming language used to manage and manipulate relational databases.
3. ASP.NET with C#: A web development framework and programming language used to build dynamic web applications.
4. ERD (Entity-Relationship Diagram): A visual representation of database schema, showing the entities and relationships between them.
5. Normalization: The process of organizing data in a database to minimize redundancy and dependency.

These are just some of the tools and techniques that can be used in this coursework. Other tools and techniques may include HTML, CSS, JavaScript, and various testing frameworks.