**Contents**

* Overview
* Objectives
* Prerequisites
* Hand-In
* Text Format
* Due Date
* Scoring
* Lab Assignment
* Views, Security and Normalization

**Overview**

**Objective**

The purpose of this lab is to become familiar with views, security and the normalization process by:

* Describing a view
* Creating, altering and dropping a views
* Retrieving data through a view
* Inserting, updating and deleting data through a view
* Creating a view with constraints
* Creating a logical model
* Creating a relation model
* Normalizing a set of attributes.

**Prerequisites**

* Read Chapter 10, pages 325 - 334
* Review the slides L17\_IntroDataModelling on D2L.
* Read the user guide L17A\_Using the SQL Developer Data Modeler on D2L.
* Review the slides L18\_Normalization on D2L.

**Demo Due Date:**

For all sections,the **lab demo** is due in 1 week (**Apr 05 – Apr 09**) by the end of your lab session.

All labs must be completed. Late labs will be marked as zero.

**Scoring:**

Lab is out of **49** marks.

View and Security – 15 marks, questions 1 to 15 are worth 1 mark each.  
 Normalization Model #1 – 10 marks   
 Data Modeling – 14 marks  
 Normalization Model #2 – 10 marks

**Lab Assignment**

**Views and Security**

* Create a table called **EMPLOYEES\_NEW** using select as from the **HR.EMPLOYEES** table. ***(1 mark)***
* Create a view called **EMPLOYEES\_VW** based on the employee numbers, employee names, and department numbers from your **EMPLOYEES\_NEW** table. The employee name should be a single column made up of the first name and last name with a space in between (e.g Steven King) Change the heading for the employee name to **EMPNAME**. ***(1 mark)***
* Display the employees who work in department 90 from the **EMPLOYEES\_VW** view.(Hint: Use **SELECT \***). ***(1 mark)***
* Select the **view name** and **text** from the **USER\_VIEWS** data dictionary view for your new **EMPLOYEES\_VW** **only**. ***(1 mark)***
* Grant select on your **EMPLOYEES\_VW** view to ICS120. ***(1 mark)***
* Create a view, based on the original **EMPLOYEES\_NEW** table, named **DEPT50\_VW** that contains the employee numbers, employee last names, and department numbers for all employees in department 50. Label the view columns **EMPNO**, **EMPLOYEE**, and **DEPTNO**. Do **not** allow an employee to be reassigned to another department through the view. (Note: Do **NOT** use the **READ ONLY** constraint to enforce this.). ***(1 mark)***
* Display the structure (e.g. **DESC**...) **and** contents (e.g. **SELECT \*** ...) of the **DEPT50\_VW** view. List only the first 10 rows when displaying the content. (Hint: Use **"rownum <= 10"**.). ***(1 mark)***
* Using the **DEPT50\_VW**, attempt to reassign the employee with a last name of **Weiss** to department **80**. Do not update the **EMPLOYEES\_NEW** table.  (If you accidentally do update the **EMPLOYEES\_NEW** table, you will need to delete the new record for Matos - employee\_ID 143 in the JOB\_HISTORY table before continuing this lab.). ***(1 mark)***
* You will get an error when you do this. Why? ***(1 mark)***
* Grant public full access to the **DEPT50\_VW** view. ***(1 mark)***
* Select the ***view name*** and ***text*** from the **USER\_VIEWS** data dictionary view to show the two views you created. ***(1 mark)***
* Using the USER\_TAB\_PRIVS\_MADE show the privileges that you have made on the **EMPLOYEES\_VW** view and the **DEPT50\_VW** view. ***(1 mark)***
* Revoke all the privileges on the **EMPLOYEES\_VW** view and the **DEPT50\_VW** view. ***(1 mark)***
* Using the USER\_TAB\_PRIVS\_MADE show that the privileges have been revoked on the **EMPLOYEES\_VW** view and the **DEPT50\_VW** view. ***(1 mark)***
* Drop all the views you created. ***(1 mark)***

**Normalization Model #1**

**Scenario:**

A client has come to you asking you to help them come up with a better way to keep track of students, courses and grades. They are currently using one big spreadsheet in Excel to keep track of everything. The spreadsheet though has become too large and they are having lots of problems maintaining it. They have provided you the following sample of the type of data in their spreadsheet.

|  |  |  |
| --- | --- | --- |
| Student Id, Name | Course Information | Grade |
| 1, John Smith | ICS112 Intro to Programming | A- |
| 1, John Smith | ICS122 Data Algorithms | C |
| 1, John Smith | MATH170 Linear Algebra | C |
| 2, Jane Doe | ICS112 Intro to Programming | B |
| 2, Jane Doe | MATH170 Linear Algebra | B+ |
| 2, Jane Doe | ENG123 Technical Writing | A+ |

Normalize this information to 1 NF, then to 2 NF, then to 3 NF. Indicate the primary keys and any foreign keys.  You can either underscore the columns or place an asterisk next to the column to indicate primary keys.   Use the following table structure to show your work.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** | **Entity** |
| Stud\_ID\* | Stud\_ID\* | Stud\_ID\* | Stud\_ID\* | Student |
| Stud\_Name | Stud\_Name | Stud\_Name | Stud\_Name |  |
| Course\_ID\* ) |  |  |  |  |
| Course\_Name ) | Stud\_ID\* | Stud\_ID\* | Stud\_ID\* | Grade |
| Grade ) | Course\_ID\* | Course\_ID\* | Course\_ID\* |  |
|  | Course\_Name | Grade | Grade |  |
|  | Grade |  |  |  |
|  |  | Course\_ID\* | Course\_ID\* | Course |
|  |  | Course\_Name | Course\_Name |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Hint:** Your model should have 3 entities.

**Marks:** – 10 marks: 3 for 1NF,3 for 2NF, 3 for 3NF and 1 for entity names.

**Data Modeling**

* Using data modeler in SQL Developer, create the logical model for the previous Normalization Model #1 question. Make sure you use proper naming conventions for entities, attributes and relationships. Save your model and demo it in SQL Developer. *(7 marks).*
* Demo generating the relational model from your logical model. *(2 marks).*
* Demo generating the DDL from your relational model and run it. *(2 marks).*
* Create and demo the SQL to insert the data into the tables you just created (3 marks)

**Normalization Model #2**

* A company has a number of salespeople who work in different company departments. A salesperson has a name, a sales commission percentage, the year of hire, and the department where they work. Salespeople sell products which have a product name and a unit price. The quantity of products sold by each salesperson is shown.



Normalize this information to 1 NF, then to 2 NF, then to 3 NF. Indicate the names of the relations and primary keys.  You can either underscore the columns or place an asterisk next to the column to indicate primary keys.   Use the following table structure to show your work.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** | **Entity** |
| Salesperson\_ID\* | Salesperson\_ID\* | Salesperson\_ID\* | Salesperson\_ID\* | Salesperson |
| Salesperson\_name | Salesperson\_name | Salesperson\_name | Salesperson\_name |  |
| Department\_ID | Department\_ID | Department\_ID | Sales\_Commission |  |
| Department\_Name | Department\_Name | Department\_Name | Hire\_Year |  |
| Sales\_Commission | Sales\_Commission | Sales\_Commission | Department\_ID (FK) |  |
| Hire\_Year | Hire\_Year | Hire\_Year |  |  |
| Product\_ID\* ) |  |  | Department\_ID\* | Department |
| Prod\_Name ) | Salesperson\_ID\* | Salesperson\_ID\* | Department\_Name |  |
| Unit\_Price ) | Product\_ID\* | Product\_ID\* |  |  |
| Quantity\_Sold ) | Prod\_Name | Quantity\_Sold | Salesperson\_ID\* | Product\_Sale |
|  | Unit\_Price |  | Product\_ID\* |  |
|  | Quantity\_Sold | Product\_ID\* | Quantity\_Sold |  |
|  |  | Prod\_Name |  |  |
|  |  | Unit\_Price | Product\_ID\* | Product |
|  |  |  | Prod\_Name |  |
|  |  |  | Unit\_Price |  |

**Hint:** 3NF has 4 entities.

**Marks:** – 10 marks: 3 for 1NF,3 for 2NF, 3 for 3NF and 1 for entity names.