MARMARA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING

CSE3055 DATABASE SYSTEMS

PROJECT – STEP 3 REPORT

Oğuzcan ÇELİK 150114062

Mehmet Ali SAKALLI 150114080

Our project is about implementing a security education company's user information to a database. We named our project as 'SECURITYLESSONS'. We have information about firms, firms' employees, lessons, instructors and instructors' cars that provided by the company.

• We have firm names and their business sector, employee names, their positions in their firms, their ages, and their grades. We have information about lessons like lesson name, how much does a lesson cost, how many times does it take for a lesson etc. Furthermore we have instructor names, their salary, their expertise and their car's models and ages.

• Each table has a primary key names after table name. There are two foreign keys in Employee table, firmID means which firm does that employee works for and lessonID means which lessons does the employee takes.

FK_EMPLOYEE_FIRM

FK_EMPLOYEE_LESSON

• On check constraints, we limited the age columns for Car, Employee and Instructor tables. By the rules of Ministry of Inferior, a car's age can not be more than 8 on a securty education company and Employee and Instructor ages restricted with 15 to 100 and 18 to 80. We also restricted Employees' grades 0 to 100. And finally a Instructor's salary should be greater than 1500. Now we're adding these check constraints names one by one:

CK_CAR_age	0 < < 8
CK_EMPLOYEE_age	15 < < 100
CK_INSTRUCTOR_age	18 < < 80
CK_EMPLOYEE_grade	0 < <= 100
CK_INSTRUCTOR_salary	1500 <

• We added 'fullName' column to the Employee table as computed column, it combines name and surname columns.

• We added firm names and lesson names as unique under 'UK_FIRM_name' and 'UK_LESSON_name' .

• We set 'place' column in Lesson table 'Sınıf' as default under 'DF_LESSON_place'.

• We indexed Instructor salaries under 'IX_INSTRUCTOR_salary'.

• We added 4 triggers in total, two for Employee table and two for Firm table.

'empInsertTrigger' track if something inserted to Employee table and if so, it prints 'Element succesfully inserted to the EMPLOYEE table.' as message and shows inserted element on Results tab.

'firmInsertTrigger' does exactly the same job for the Firm table.

‘empDeleteTrigger’ tracks if an element removed from the Employee table and prints 'Element(s) succesfully deleted from the EMPLOYEE table.’ as message.

‘firmDeleteTrigger’ does exactly the same job fort the Firm table.

• We have 6 store procedures in our database:

‘spDeleteByEmployeeId’ deletes the element which matches with the Employee id given as paramater.

‘spGetEmployeesByFirmID’ lists the all employess of given firm id.

‘spInsertEmployee’ inserts new employee to the table. Employee attributes should be given as parameter.

‘spListInstructorsBySalary’ lists instructors ordered by their salary in descending order.

‘spListInstructorsCars’ lists each instructor’s car model and car’s age.

‘spUpdateLessonCostByLessonId’ updates the cost of the lesson. Lesson id should be given as parameter. Then shows the current value of that element.

• We have 4 view in our database:

‘vWAverageTimeOfLessonByFirmId’ computes average time of each firm’s employees’ lesson time.

‘vWNumberOfCarsByInstructors’ computes the instructor number for each car model.

‘vWNumberOfEmployeesByFirm’ computes the number of employees for each firm.

‘vWNumberOfStudentsBYLesson’ computes total student number of each lesson.

UPDATED ER DIAGRAM