Readings:

* chapter 22: “Business Intelligence: An Introduction” (Petkovic)
* chapter 9: “Data Warehousing” (Hoffer, Ramesh, & Topi) from page 375 - 394

Homework:

* chapter 9: “Data Warehousing” (Hoffer, Ramesh, & Topi)
  + Problems and Exercises 1, 2, 3

1. Examine the three tables with student data shown in Figure 9-1. Design a single-table format that will hold all of the data (nonredundantly) that are contained in these three tables. Choose column names that you believe are most appropriate for these data.

**StudentNo, LastName, MI, FirstName, Telephone, Status, Address, Dept, Hours, Insurance**

2. Your assignment involves two parts:

* 1. Construct tables for 06/21/2010 and 06/22/2010, reflecting these transactions; assume that the data are transient (refer to Figure 9-7).

**06/20/2010:**

**Key Name Major**

**001  Amy Music**

**002  Tom Business**

**003  Sue Art**

**004  Joe Math**

**005  Ann Engineering**

**06/21/2010:**

**Key Name Major**

**001  Amy Music**

**002  Tom Business**

**003  Sue Art**

**004  Joe Business**

**006  Jim Phys Ed**

**06/22/2010:**

**Key Name Major**

**001  Amy Music**

**002  Tom Business**

**003  Sue History**

**004  Joe Business**

**006  Jim Basket Weaving**

* 1. Construct tables for 06/21/2010 and 06/22/2010, reflecting these transactions; assume that the data are periodic (refer to Figure 9-8).

**06/20/2010:**

**Key Date Name Major Action**

**001  06/20 Amy Music C**

**002  06/20 Tom Business C**

**003 06/20 Sue Art C**

**004 06/20 Joe Math C**

**005 06/20 Ann Engineering C**

**06/21/2010:**

**Key Date Name Major Action**

**001  06/20 Amy Music C**

**002  06/20 Tom Business C**

**003 06/20 Sue Art C**

**004 06/20 Joe Math C**

**004 06/21 Joe Business U**

**005 06/20 Ann Engineering C**

**005 06/21 Ann Engineering D**

**006 06/21 Jim Phys Ed C**

**06/22/2010:**

**Key Date Name Major Action**

**001  06/20 Amy Music C**

**002  06/20 Tom Business C**

**003 06/20 Sue Art C**

**003 06/22 Sue History U**

**004 06/20 Joe Math C**

**004 06/21 Joe Business U**

**005 06/20 Ann Engineering C**

**005 06/21 Ann Engineering D**

**006 06/21 Jim Phys Ed C**

**006 06/22 Jim Basket Weaving U**

3. The only fact that is to be recorded in the fact table is CourseGrade.

1. Design a star schema for this problem. See Figure 9-10 for the format you should follow.

Professor :

ProfID

ProfName

Title

DepartmentID DepartmentName

CourseSection:

CourseID

SectionNumber

CourseName

Units

RoomID

RoomCapacity

CourseGrade:

CourseID

SemesterID

ProfID

StudentID

AverageScore

SDScore

Period: SemesterID

Year

Student : StudentID

StudentName Major

1. Estimate the number of rows in the fact table, using the assumptions stated previously.

**930.**

1. Estimate the total size of the fact table (in bytes), assuming that each field has an average of 5 bytes.

**4650B.**

1. If you didn’t want to or didn’t have to stick with a strict star schema for this data mart, how would you change the design? Why?

**Combine the tables as much as possible to avoid redundant data and save space.**

1. Various characteristics of sections, professors, and students change over time. How do you propose designing the star schema to allow for these changes? Why?

**Record the average, range, and standard deviation instead of the actual data to avoid the great change.**

* Using the sample database from prior lecture, write the following persistent stored modules, and write commands to execute each:

1. Create a stored procedure that takes Country as an input parameter, and outputs a list of employees in that country.
   1. Procedure Name: GetEmployees @Country.
   2. Columns: First Name, Last Name, City, Country

create procedure GetEmployees @Country varchar(10)

as

begin

if @Country = 'England'

begin

select e.[emp\_fname], e.[emp\_lname], d.[Location] As City, @Country as country

from employee e join department d on e.dept\_no = d.dept\_no

where d.location = 'London'

end

if @Country = 'USA'

begin

select e.[emp\_fname], e.[emp\_lname], d.[Location] As City, @Country as country

from employee e join department d on e.dept\_no = d.dept\_no

where d.location <> 'London'

end

end

GetEmployees 'England'

A screenshot of a cell phone

Description automatically generated

GetEmployees 'USA'

A screenshot of a cell phone

Description automatically generated

1. Write a stored procedure AssignWork which takes emp\_no, project\_no, job as input parameters. It must add a record to the works\_on table with enterd\_date = system date, and returns the employee full name as an output parameter

create procedure AssignWork @emp\_no int, @project\_no char(4), @job varchar(50)

as

declare @FullName varchar(50)

begin

insert into works\_on values(@emp\_no,@project\_no,@job,GETDATE())

select @FullName = [emp\_fname] + ' ' + [emp\_lname]

from employee

where emp\_no = @emp\_no

select @FullName as FullName

end

exec AssignWork 15000,'p3','Manager'

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create a UDF that takes @City as an input parameter and returns country.

Procedure Name: GetCountry (@city)

create function GetCountry

(@City varchar(25))

returns varchar(10)

as

begin

declare @Country varchar(10)

if @City = 'London'

begin

set @Country = 'England'

end

else

begin

set @Country = 'USA'

end

return @Country

end

select dbo.GetCountry ('London') as country

A screenshot of a cell phone

Description automatically generated

select dbo.GetCountry ('Boston') as country

A screenshot of a cell phone

Description automatically generated