**CMS Project: Phase II Instructions**

In this phase, you will create tables based upon the ERD and SQL code below. You will then populate each table with the data presented below. Finally, you will create queries that will be used to support reports for Accounting and Management. You will not actually create the reports in a GUI environment– only the queries that will serve as the basis for the reports. Screenshots are required for a grade to be given. One screenshot is not the idea; however, multiple screenshots along the way is the goal.

**Background**:

The following ERD will be used as the basis for this Phase.



**Part A: Table Creation and Data Loading**

**Instructions**: Create a new database in SQL Server and run the following CREATE TABLE commands. Note that you must run the CREATE TABLE statements in the order presented (and load the data in the order presented) to avoid conflicts resulting from foreign key constraints.

**Additional instructions for materials to turn in for this phase of your project are included at the end of this specification document.**

CREATE TABLE Regions

(RegionID int not null,

RegionAbbreviation varchar(4),

RegionName varchar(100),

CONSTRAINT PK\_Regions PRIMARY KEY (RegionID))

CREATE TABLE Countries

(CountryID int not null,

CountryName varchar(50),

WeeklyHours int,

Holidays int,

VacationDays int,

RegionID int,

CONSTRAINT PK\_Countries PRIMARY KEY (CountryID),

CONSTRAINT FK\_CountriesRegions FOREIGN KEY (RegionID) References Regions)

CREATE TABLE EmployeeTitles

(TitleID int not null,

Title varchar(15),

CONSTRAINT PK\_EmpTitles PRIMARY KEY (TitleID))

CREATE TABLE BillingRates

(TitleID int not null,

Level int not null,

Rate float,

CurrencyName varchar(5),

CONSTRAINT PK\_BillingRates PRIMARY KEY (TitleID, Level),

CONSTRAINT FK\_BillingRatesTitles FOREIGN KEY (TitleID) References EmployeeTitles)

CREATE TABLE Employees

(EmpID int not null,

FirstName varchar(30),

LastName varchar(30),

Email varchar(50),

Salary decimal(10,2),

TitleID int,

Level int,

SupervisorID int,

CountryID int,

CONSTRAINT PK\_Employees PRIMARY KEY (EmpID),

CONSTRAINT FK\_EmployeesCountries FOREIGN KEY (CountryID) References Countries,

CONSTRAINT FK\_EmployeesEmpTitles FOREIGN KEY (TitleID) References EmployeeTitles,

CONSTRAINT FK\_EmployeeSupervisors FOREIGN KEY (SupervisorID) References Employees)

CREATE TABLE ContactTypes

(ContactTypeID int not null,

ContactType varchar(30)

CONSTRAINT PK\_ContactTypes PRIMARY KEY (ContactTypeID))

CREATE TABLE ContractTypes

(ContractTypeID int not null,

ContractType varchar(30)

CONSTRAINT PK\_ContractTypes PRIMARY KEY (ContractTypeID))

CREATE TABLE BenefitTypes

(BenefitTypeID int not null,

BenefitType varchar(30)

CONSTRAINT PK\_BenefitTypes PRIMARY KEY (BenefitTypeID))

CREATE TABLE Clients

(ClientID int not null,

LegalName varchar(50),

CommonName varchar(50),

AddrLine1 varchar(50),

AddrLine2 varchar(50),

City varchar(25),

State\_Province varchar(25),

Zip varchar(9),

CountryID int,

CONSTRAINT PK\_Clients PRIMARY KEY (ClientID),

CONSTRAINT FK\_ClientsCountries FOREIGN KEY (CountryID) REFERENCES Countries)

CREATE TABLE Contacts

(ContactID int not null,

FirstName varchar(50),

LastName varchar(50),

AddrLine1 varchar(50),

AddrLine2 varchar(50),

City varchar(25),

State\_Province varchar(25),

Zip varchar(9),

CountryID int,

ContactTypeID int,

CONSTRAINT PK\_Contacts PRIMARY KEY (ContactID),

CONSTRAINT FK\_ContactsCountries FOREIGN KEY (CountryID) REFERENCES Countries)

CREATE TABLE ContractTypes

(ContractTypeID int not null,

ContractTypeDesc varchar(50),

CONSTRAINT PK\_ContractTypes PRIMARY KEY (ContractTypeID))

CREATE TABLE Contracts

(ContractID int not null,

ContractDesc varchar(100),

ClientID int,

ContractTypeID int,

CONSTRAINT PK\_Contracts PRIMARY KEY (ContractID),

CONSTRAINT FK\_ContractsClients FOREIGN KEY (ClientID) REFERENCES Clients,

CONSTRAINT FK\_ContractsContractTypes FOREIGN KEY (ContractTypeID) REFERENCES ContractTypes)

CREATE TABLE ContractsContacts

(ContractID int not null,

ContactID int not null,

CONSTRAINT PK\_ContractsContacts PRIMARY KEY (ContractID, ContactID),

CONSTRAINT FK\_CC\_Contracts FOREIGN KEY (ContractID) REFERENCES Contracts,

CONSTRAINT FK\_CC\_Contacts FOREIGN KEY (ContactID) REFERENCES Contacts)

CREATE TABLE Projects

(ProjectID int not null,

ProjectName varchar(50),

HourCapAmount decimal(10,2),

ProjectManagerID int,

ContractID int,

CONSTRAINT PK\_Projects PRIMARY KEY (ProjectID),

CONSTRAINT FK\_ProjectsEmployees FOREIGN KEY (ProjectManagerID) REFERENCES Employees,

CONSTRAINT FK\_ProjectsContracts FOREIGN KEY (ContractID) REFERENCES Contracts)

CREATE TABLE EmployeesProjects

(EmpID int not null,

ProjectID int not null,

StartDate smalldatetime,

EndDate smalldatetime,

CONSTRAINT PK\_EmployeesProjects PRIMARY KEY (EmpID, ProjectID),

CONSTRAINT FK\_EP\_Employees FOREIGN KEY (EmpID) REFERENCES Employees,

CONSTRAINT FK\_EP\_Projects FOREIGN KEY (ProjectID) REFERENCES Projects)

CREATE TABLE Timesheets

(TimesheetID int not null,

SupervisorApproveDate smalldatetime,

CONSTRAINT PK\_Timesheets PRIMARY KEY (TimesheetID))

CREATE TABLE WorkHours

(EmpID int not null,

ProjectID int not null,

WH\_Day int not null,

WH\_Month int not null,

WH\_Year int not null,

HoursWorked float,

TimesheetID int,

CONSTRAINT PK\_WorkHours PRIMARY KEY (EmpID, ProjectID, WH\_Day, WH\_Month, WH\_Year),

CONSTRAINT FK\_WorkHoursEmployees FOREIGN KEY (EmpID) REFERENCES Employees,

CONSTRAINT FK\_WorkHoursProjects FOREIGN KEY (ProjectID) REFERENCES Projects,

CONSTRAINT FK\_WorkHoursTimesheets FOREIGN KEY (TimesheetID) REFERENCES Timesheets)

CREATE TABLE BenefitsTaken

(EmpID int not null,

BenefitTypeID int not null,

BT\_Day int not null,

BT\_Month int not null,

BT\_Year int not null,

HoursTaken float,

TimesheetID int,

CONSTRAINT PK\_BenefitsTaken PRIMARY KEY (EmpID, BenefitTypeID, BT\_Day, BT\_Month, BT\_Year),

CONSTRAINT FK\_BenefitsTakenEmployees FOREIGN KEY (EmpID) REFERENCES Employees,

CONSTRAINT FK\_BenefitsTakenBenefitTypes FOREIGN KEY (BenefitTypeID) REFERENCES BenefitTypes,

CONSTRAINT FK\_BenefitsTakenTimesheets FOREIGN KEY (TimesheetID) REFERENCES Timesheets)

**Data Section**

The following information is currently maintained in various spreadsheets throughout CMS. Data from these spreadsheets must be uploaded into your newly created tables before the database can be considered operational.

**REGIONS**

ID Abbr. Region Name

1 NAR North America

2 CALA Central and Latin America

3 APAC Asia and Pacific

4 EMEA Europe, Middle East, and Africa

**COUNTRIES**

ID Country Name Weekly Hours Holidays Vacation Days Region

1 United States 40 11 10 NAR

2 Canada 40 12 15 NAR

3 United Kingdom 38 10 10 EMEA

4 France 38 14 10 EMEA

5 Ireland 38 10 15 EMEA

6 Italy 35 9 20 EMEA

7 Thailand 40 17 20 APAC

8 Singapore 40 17 21 APAC

9 Panama 40 12 15 CALA

**BENEFIT TYPES**

ID Benefit Type Name

1 Vacation

2 Holiday

3 Jury Duty

4 Maternity Leave

5 Paternity Leave

6 Military Duty

**CONTACT TYPES**

ID Contact Type Name

1 Systems Engineer

2 Sales

3 Billing

**CONTRACT TYPES**

ID Contract Type Name

1 Maintenance

2 Fixed Price

3 License

4 Time and Materials

**CLIENTS**

ID Legal Name Common Address1 Address2 City State Zip Country

1 BMA British Mobile 130 Wake Dr. Wake NC 24539 US

2 FT France Mobile 123 East St. Suite #2 Paris 45678 France

3 IBC IBC 456 Main Johor 78945 Singapore

4 MTM MTM 6789 First St. Mead GA 45678 US

5 BT Britain Tele 98769 Park St. Level 3 London 48695 UK

**CONTRACTS**

ID ContractDesc Contract Type Client

1 Work Order 1 Maint FT

2 Work Order 1 T&M BT

3 Work Order 1 Fixed Price IBC

4 Work Order 2 Maint IBC

5 Work Order 1 Fixed Price MTM

6 Work Order 2 T&M FT

**CONTACTS**

ID First Last Addr1 Addr2 City State Zip Country Type

1 Bugg Bunny 123 Looney NoWhere AK 45678 US SysEng

2 Elmer Fudd 789 Park Pl. Apt 3 Skyville NM 45678 US Billing

3 Daffy Duck 45678 One St. Norwood 45678 UK Sales

4 Darth Vader 456 Two St. Towns 47896 UK Sales

5 Luke Sky #4 Tatooine Paris 45678 France Billing

6 Princess Lea 723 Coruscant Rome 45678 Italy SysEng

7 John Doe 987 Main St. Paris 78945 France SysEng

8 Jane Doe 7658 Oak Ln. Crue VA 45678 US SysEng

**CONTRACTS’ CONTACTS**

Contract Client Contact Name

Work Order 1 BT Daffy Duck

Work Order 1 FT John Doe, Jane Doe, Princess Lea

Work Order 2 FT Elmer Fudd

Work Order 1 IBC Buggs Bunny

Work Order2 IBC Luke Sky

Work Order 2 IBC Darth Vader

Work Order 1 MTM Daffy Duck

**EMPLOYEE TITLES**

ID Title

1 Consultant

2 Analyst

3 Director

**BILLING RATES**

TitleID Level Rate Currency

1 1 150.00 USD

1 2 200.00 USD

1 3 300.00 USD

2 1 50.00 USD

2 2 100.00 USD

2 3 150.00 USD

3 1 250.00 USD

3 2 350.00 USD

3 3 450.00 USD

**EMPLOYEES**

ID First Last CountryID Email Salary Title LevelID

1 Matthew Smith 1 [msmith@cms.com](mailto:msmith@cms.com) 45000 Consultant 1

2 Mark Jones 1 [mjones@cms.com](mailto:mjones@cms.com) 94000 Director 1

3 Luke Rice 4 [lrice@cms.com](mailto:lrice@cms.com) 65000 Consultant 2

4 John Rich 5 [jrich@cms.com](mailto:jrich@cms.com) 74000 Consultant 3

5 James Doe 6 [jDoe@cms.com](mailto:jbolt@cms.com) 40000 Analyst 1

6 Peter Pride 3 [ppride@cms.com](mailto:ppride@cms.com) 60000 Analyst 2

7 Eric Potter 3 [epotter@cms.com](mailto:epotter@cms.com) 81000 Consultant 3

8 Paul Davis 1 [pdavis@cms.com](mailto:pdavis@cms.com) 103000 Director 2

**PROJECTS**

ID Project Name HourCapAmount ProjectManager Contracts Client

1 IBC – India 120 Davis Work Order 2 IBC

2 FT-Maint 100 Doe Work Order 2 FT

3 BT – WO 1 Time 270 Rich Work Order1 BT

4 BT – WO1 Materials Rich Work Order1 BT

5 IBC - WO1 Davis Work Order 1 IBC

6 IBC – WO2 Davis Work Order 2 IBC

7 MTM – WO1 Pride Work Order 1 MTM

8 FT – WO2 Time 500 Doe Work Order 2 FT

9 FT –WO2 Materials Doe Work Order 2 FT

**PROJECT-EMPOYEE ASSIGNMENTS**

Employee Project StartDate EndDate

Doe IBC-India 1/1/2013

Doe IBC - WO1 5/7/2013

Doe BT – WO1 Materials 2/1/2013 4/30/2013

Smith FT-Maint 2/1/2013

Jones FT-Maint 3/1/2013

Rice MTM – WO1 1/1/2013

**WORK HOURS**

Employee Project Day Month Year HoursWorked TimeSheet

Doe IBC-India 2 4 2013 8 1

Doe IBC-India 3 4 2013 8 1

Doe IBC-India 4 4 2013 8 1

Doe IBC-India 5 4 2013 8 1

Doe IBC-India 6 4 2013 8 1

Doe IBC-India 9 4 2013 8 1

Doe IBC-India 10 4 2013 8 1

Doe IBC-India 11 4 2013 8 1

Doe IBC-India 12 4 2013 8 1

Doe IBC-India 13 4 2013 4 1

Doe IBC - WO1 13 4 2013 4 1

Doe IBC - WO1 16 4 2013 4 1

Doe IBC - WO1 16 4 2013 4 1

Doe IBC-India 17 4 2013 8 1

Doe IBC-India 18 4 2013 8 1

Doe IBC-India 19 4 2013 5 1

Doe IBC-WO1 19 4 2013 3 1

Doe IBC-India 20 4 2013 8 1

Doe IBC-India 23 4 2013 8 1

Doe IBC-India 24 4 2013 8 1

Doe IBC-India 26 4 2013 8 1

Doe IBC-India 27 4 2013 8 1

Doe IBC-India 30 4 2013 8 1

Doe IBC-WO1 1 5 2013 8 2

Doe IBC-WO1 2 5 2013 8 2

Doe IBC-WO1 3 5 2013 8 2

Doe IBC-WO1 4 5 2013 8 2

Doe IBC-India 7 5 2013 8 2

Doe IBC-WO1 8 5 2013 8 2

Doe IBC-WO1 9 5 2013 8 2

Doe IBC-WO1 10 5 2013 8 2

Doe IBC-WO1 11 5 2013 8 2

Doe IBC-India 14 5 2013 8 2

Doe IBC-WO1 15 5 2013 8 2

Doe IBC-WO1 16 5 2013 8 2

Doe IBC-WO1 17 5 2013 8 2

Doe IBC-WO1 18 5 2013 8 2

Doe IBC-India 21 5 2013 8 2

Doe IBC-WO1 22 5 2013 8 2

Doe IBC-WO1 23 5 2013 8 2

Doe IBC-WO1 24 5 2013 8 2

Doe IBC-India 28 5 2013 8 2

Doe IBC-WO1 29 5 2013 8 2

Doe IBC-WO1 30 5 2013 8 2

Doe IBC-WO1 31 5 2013 8 2

Jones FT-Maint 2 4 2013 8 3

Jones FT-Maint 3 4 2013 8 3

Jones FT-Maint 4 4 2013 8 3

Jones FT-Maint 5 4 2013 8 3

Jones FT-Maint 6 4 2013 8 3

Jones FT-Maint 9 4 2013 8 3

Jones FT-Maint 10 4 2013 8 3

Jones FT-Maint 11 4 2013 8 3

Jones FT-Maint 12 4 2013 8 3

Jones FT-Maint 13 4 2013 15 3

Jones FT-Maint 16 4 2013 14 3

Jones FT-Maint 17 4 2013 8 3

Jones FT-Maint 18 4 2013 8 3

Jones FT-Maint 19 4 2013 10 3

Jones FT-Maint 20 4 2013 8 3

Jones FT-Maint 23 4 2013 8 3

Jones FT-Maint 24 4 2013 8 3

Jones FT-Maint 26 4 2013 8 3

Jones FT-Maint 27 4 2013 8 3

Jones FT-Maint 30 4 2013 8 3

Jones FT-Maint 1 5 2013 8 4

Jones FT-Maint 2 5 2013 8 4

Jones FT-Maint 3 5 2013 8 4

Jones FT-Maint 4 5 2013 8 4

Jones FT-Maint 7 5 2013 8 4

Jones FT-Maint 8 5 2013 8 4

Jones FT-Maint 9 5 2013 8 4

Jones FT-Maint 10 5 2013 8 4

Jones FT-Maint 11 5 2013 8 4

Jones FT-Maint 14 5 2013 8 4

Jones FT-Maint 15 5 2013 8 4

Jones FT-Maint 16 5 2013 8 4

Jones FT-Maint 17 5 2013 8 4

Jones FT-Maint 18 5 2013 8 4

Jones FT-Maint 21 5 2013 8 4

Jones FT-Maint 22 5 2013 8 4

Jones FT-Maint 28 5 2013 8 4

Jones FT-Maint 29 5 2013 8 4

Jones FT-Maint 30 5 2013 8 4

Jones FT-Maint 31 5 2013 8 4

Smith FT-Maint 2 4 2013 8 5

Smith FT-Maint 3 4 2013 8 5

Smith FT-Maint 4 4 2013 8 5

Smith FT-Maint 5 4 2013 8 5

Smith FT-Maint 9 4 2013 8 5

Smith FT-Maint 10 4 2013 8 5

Smith FT-Maint 11 4 2013 8 5

Smith FT-Maint 12 4 2013 8 5

Smith FT-Maint 16 4 2013 14 5

Smith FT-Maint 17 4 2013 8 5

Smith FT-Maint 18 4 2013 8 5

Smith FT-Maint 19 4 2013 10 5

Smith FT-Maint 20 4 2013 8 5

Smith FT-Maint 23 4 2013 8 5

Smith FT-Maint 24 4 2013 8 5

Smith FT-Maint 26 4 2013 8 5

Smith FT-Maint 27 4 2013 8 5

Smith FT-Maint 30 4 2013 8 5

Smith FT-Maint 1 5 2013 8 6

Smith FT-Maint 2 5 2013 8 6

Smith FT-Maint 3 5 2013 8 6

Smith FT-Maint 4 5 2013 8 6

Smith FT-Maint 7 5 2013 8 6

Smith FT-Maint 8 5 2013 8 6

Smith FT-Maint 9 5 2013 8 6

Smith FT-Maint 10 5 2013 8 6

Smith FT-Maint 11 5 2013 8 6

Smith FT-Maint 14 5 2013 8 6

Smith FT-Maint 15 5 2013 8 6

Rice MTM – WO1 2 4 2013 8 7

Rice MTM – WO1 3 4 2013 8 7

Rice MTM – WO1 4 4 2013 8 7

Rice MTM – WO1 5 4 2013 8 7

Rice MTM – WO1 6 4 2013 8 7

Rice MTM – WO1 9 4 2013 8 7

Rice MTM – WO1 10 4 2013 8 7

Rice MTM – WO1 11 4 2013 8 7

Rice MTM – WO1 12 4 2013 8 7

Rice MTM – WO1 16 4 2013 14 7

Rice MTM – WO1 17 4 2013 8 7

Rice MTM – WO1 18 4 2013 8 7

Rice MTM – WO1 19 4 2013 10 7

Rice MTM – WO1 20 4 2013 8 7

Rice MTM – WO1 23 4 2013 8 7

Rice MTM – WO1 24 4 2013 8 7

Rice MTM – WO1 26 4 2013 8 7

Rice MTM – WO1 27 4 2013 8 7

Rice MTM – WO1 30 4 2013 8 7

Rice MTM – WO1 1 5 2013 8 8

Rice MTM – WO1 2 5 2013 8 8

Rice MTM – WO1 3 5 2013 8 8

Rice MTM – WO1 4 5 2013 8 8

Rice MTM – WO1 7 5 2013 8 8

Rice MTM – WO1 8 5 2013 8 8

Rice MTM – WO1 9 5 2013 8 8

Rice MTM – WO1 10 5 2013 8 8

Rice MTM – WO1 11 5 2013 8 8

Rice MTM – WO1 14 5 2013 8 8

Rice MTM – WO1 15 5 2013 8 8

**BENEFITS TAKEN**

Employee Day Month Year BenefitType TimeSheet

Doe 25 4 2013 Holiday 1

Doe 25 5 2013 Holiday 2

Jones 25 4 2013 Holiday 3

Jones 23 5 2013 Vacation 3

Jones 24 5 2013 Vacation 4

Jones 25 5 2013 Holiday 4

Smith 6 4 2013 Vacation 5

Smith 25 4 2013 Holiday 5

Smith 25 5 2013 Holiday 6

Rice 25 4 2013 Holiday 7

Rice 25 5 2013 Holiday 8

**TIMESHEETS**

ID SupervisorApproveDate

1 4/30/2013

2 5/31/2013

3 4/30/2013

4 5/31/2013

5 4/30/2013

6 5/31/2013

7 4/30/2013

8 5/31/2013

**Part B: Reports**

1. **Human Resources:**

The HR Department requires a list of all the employees who are employed by CMS. This information should be organized as follows:

|  |
| --- |
| Region |
| Country |
| Employee name (Last, First) |
| Title + Level (e.g. “Consultant - 1”) |
| Salary (in USD) |

**\*\*Sort data in ascending order first by Region, then by Country, then by Employee last name, then by Title, then by Salary**

**Instructions:**

For this assignment, write the query that produces the results as described above.

1. **Invoicing**

Accounting requires information to produce invoices. For each client, CMS’s Invoicing Controller must know the following information as of the last day of each month:

|  |
| --- |
| Client Name |
| Contract Name(s) |
| Project(s) |
| Employees who logged hours to a project from the first day of the current month until the last day of the current month |
| Total number of hours logged for each employee during the month |
| Employee rate |
| Total Charges per employee (i.e. employee rate x employee hours worked) |
| Billing contact(s) [name, address] for each contract |

**\*\*Sort data in ascending order first by Client, then by Project, then by employee.**

**Instructions:**

All of this information should be produced using a single query that can serve as the basis for a report. Do not use views or stored procedures in conjunction with your query.

For this assignment, **you will write your query for only the month of April 2013**. You may hardcode the month number in your query. In the real-world, you would likely run this report for the current month, in which case you would want to use the getdate() function to retrieve the current date. Conversely, you might produce this query as a stored procedure that takes a given month as an argument and returns a resultset. For your assignment, however, just assume this report will be run for April 2013 and hardcode this date in your query to produce the results.

1. **Benefit Tracking**

The Human Resources department requires a report that provides information on benefit information. Assume a calendar year whereby new benefit allotments are granted as of January 1 and must be used by December 31 of same year. No carryover benefits are allowed.

|  |
| --- |
| Number of benefits days allotted to each employee |
| Number of benefit days taken year-to-date |
| Number of benefit days remaining in the calendar year |
| Number of holidays allotted to each employee |
| Number of holidays taken year-to-date |
| Number of holidays remaining in the calendar year |

**\*\*Data should be sorted in ascending order by Employee Last Name**

**Instructions:**

For this assignment, write a query that produces the results described above. **Assume that you are running the report for the 2013 calendar year**. As in the previous report, in the real-world, you would likely use the getdate() function to determine the current date and run the report from the beginning of the current year until the present time. For this assignment, however, you may hardcode the year 2013 in your query and retrieve all of the data for that year.

1. **Management Exception reporting**
2. Management must keep track of employees whose combined hours have exceeded the maximum allowed hours on projects. This report must be run before invoicing occurs in order to prevent billing in excess of contractual amounts. Show only projects whose cap amounts have been exceeded.

|  |
| --- |
| Project Name |
| Maximum allowed hours per project |
| Total hours worked on project |
| Overage (the difference between the cap and actual hours) |

**\*\*Sort data by Project Name**

1. In a separate query, show the details for the projects whose cap amounts have been exceeded:

|  |
| --- |
| Project Name |
| Employees who worked on project |
| Total hours worked on project per employee |

**\*\*Sort data by Project Name, then by employees who worked on the project**

**Instructions:**

For this assignment, write a query for 4(a) and a separate query for 4(b). The results should reflect the requirements described above.

1. **Payroll**

The Payroll department requires a report of employees who are logging more hours per week than they are legally required to work per country stipulations. These employees are paid overtime wages for hours worked in excess of weekly stipulated hours.

|  |
| --- |
| Employee Name |
| Employee Country |
| Weekly Hours per employee per country |
| Hours logged by employee in current week |

**Instructions:**

For this assignment, produce a query that determines employees who have incurred overtime during April 2013.

**Phase II Deliverables:**

1. In a Word document, take screen shots of the data in each of your tables using basic SELECT statements.

For example, SELECT \* from Clients

1. Write queries for each of the reports above. In the same Word document, include screenshots of your queries from SQL Server Express (or SQL Server). Below EACH query, include (via screen shots) the results of each query.
2. Name your Word document as follows:

Phase II CMS Project – [your last name followed by your first initial]

Using the link provided in Blackboard, upload the CMS Project: Phase II by 11:59 p.m. (ET) on **Friday** of Module/Week 8.