/\*I actually did the views for a normalization where Activity and ProjectActivity are one, since the 21st after class chris and I ciro asked the professor after class corrected that elsewise there would be a dependency\*/

/\*should we correct to a 3rd version of the tables where activity and project activity are one?

seem to have made these views for a ER without ProjectActivity where an Activity would directly have both activityId and project Id as Primary keys rather than making that an extra table\*/

go

CREATE VIEW vw\_Project AS

SELECT project.projectId, projectName, firm.firmFedID, firmName, firmAddress, fundedbudget, startDate, status, projectTypeCode, projectTypeDesc, projectedEndDate, projectManager

FROM Project

inner join Firm

on project.firmFedID = firm.firmFedID

inner join ProjectType

on project.projectId = ProjectType.projectID

/\* drop view vw\_Project

select \* from vw\_Project \*/

go

Create view vw\_activity AS

SELECT Activity.activityId, Activity.activityName, Activity.projectId,

Activity.costToDate, Activity.status, ActivityType.activityTypeCode,

ActivityType.activityTypeDesc, Activity.startDate, Activity.endDate

FROM Activity inner join ActivityType

on activity.activityId = activityType.activityID

/\* select \* from vw\_activity \*/

go

CREATE VIEW vw\_Employee AS

SELECT empNumber, firstName, lastName, ssn, address, state, zip, job

FROM Employee

/\* SELECT \* FROM vw\_employee \*/

GO

CREATE VIEW vw\_LateProjects AS

SELECT ProjectName, ActivityName, endDate, projectedEndDate

FROM Project, Activity

WHERE project.projectID = activity.projectID AND Activity.endDate > project.projectedEndDate

AND project.status = 'Active'

/\* SELECT \* from vw\_lateProjects \*/

GO

CREATE VIEW vs\_OverBudget AS

select ProjectName, sum(Activity.costToDate) AS totalCosts, fundedbudget

from project, Activity

where project.projectID = activity.projectID

group by ProjectName, FundedBudget

having sum(costToDate) >= fundedbudget

/\* select \* from vs\_OverBudget \*/

GO

CREATE VIEW vw\_OverAllocatedPM AS

SELECT Employee.firstName, Employee.lastName, Employee.ssn, COUNT(project.projectManager) as AmountofProjects

FROM Employee inner join project

on employee.empNumber = project.projectManager

GROUP BY project.projectManager, employee.FirstName, employee.LastName, employee.Ssn

having COUNT(project.projectManager) > 6

/\* select \* from vw\_OverAllocatedPM \*/

GO

CREATE VIEW vw\_CostlyActivity AS

SELECT Project.projectId, Project.projectName, Activity.activityName, Activity.costToDate

FROM Project

INNER JOIN Activity

ON project.projectID = activity.projectID

WHERE Activity.status IN ('Cancelled','On-Hold' ) AND Project.status = 'Active'

/\* select \* from vw\_CostlyActivity \*/

GO

CREATE VIEW vw\_CompletedProjects AS

SELECT Project.projectId, Project.projectName, ProjectType.projectTypeDesc

FROM Project

INNER JOIN Activity

ON activity.projectID = project.projectID

INNER JOIN ProjectType

ON project.projectID = projectType.projectID

INNER JOIN ActivityType

ON activity.activityID = activityType.activityID

WHERE ActivityType.activityTypeDesc = 'WARRANTY' AND Activity.endDate IS NOT NULL

/\* select \* from vw\_CompletedProjects \*/

GO

CREATE VIEW vw\_FundedProjectsNotStarted AS

SELECT project.projectId, project.projectName, project.projectedEndDate

FROM

project

LEFT JOIN activity

ON activity.projectId = project.projectId

WHERE

Activity.projectId IS NULL

AND project.projectId IS NOT NULL

/\* select \* from ww\_FundedProjectsNotStarted \*/

GO