```sql
-- Created by Vertabelo (http://vertabelo.com)
-- Last modification date: 2015-12-09 02:14:00.563




-- tables
-- Table: Address
CREATE TABLE vamseepr2.Address (
    AddressId   INTEGER        NOT NULL    IDENTITY(1, 1),
    Address1    NVARCHAR(30)   NOT NULL,
    Address2    NVARCHAR(30)   NULL,
    City        NVARCHAR(30)   NOT NULL,
    State       NVARCHAR(50)   NOT NULL,
    Zipcode     NVARCHAR(5)    NOT NULL    CHECK (Zipcode LIKE '[1-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT  AddressPk      PRIMARY KEY (AddressId)
)
;

-- Table: Benefits
CREATE TABLE vamseepr2.Benefits (
    BenefitId       INTEGER        NOT NULL    IDENTITY(1, 1),
    BenefitCost     DECIMAL(5,2)   NOT NULL    CHECK (BenefitCost >= 0.0),
    SelectionType   INTEGER        NOT NULL,
    Description     NVARCHAR(100)  NULL,
    CONSTRAINT      BenefitsPk     PRIMARY KEY (BenefitId)
)
;

-- Table: Buildings
CREATE TABLE vamseepr2.Buildings (
    BuildingId      INTEGER        NOT NULL    IDENTITY(1, 1),
    BuildingName    NVARCHAR(50)   NOT NULL,
    CONSTRAINT      BuildingsPk    PRIMARY KEY (BuildingId)
)
;

-- Table: Classroom
CREATE TABLE vamseepr2.Classroom (
    ClassroomId         INTEGER        NOT NULL    IDENTITY(1, 1),
    BuildingId          INTEGER        NOT NULL,
    RoomNumber          NUMERIC(3,0)   NOT NULL    CHECK (RoomNumber > 0),
    MaxSeating          NUMERIC(3,0)   NOT NULL    CHECK (MaxSeating > 0),
    WhiteBoardCount     NUMERIC(2,0)   NULL        CHECK (WhiteBoardCount >= 0),
    AVEquip             NVARCHAR(1000) NULL,
    ProjectorId         INTEGER        NOT NULL,
    CONSTRAINT          ClassroomPk    PRIMARY KEY (ClassroomId)
)
;

-- Table: College
CREATE TABLE vamseepr2.College (
    CollegeId       INTEGER        NOT NULL    IDENTITY(1, 1),
    CollegeName     NVARCHAR(150)  NOT NULL,
    CONSTRAINT      CollegePk      PRIMARY KEY (CollegeId)
)
;

-- Table: CourseDailySchedule
CREATE TABLE vamseepr2.CourseDailySchedule (
    DailyID             INTEGER    NOT NULL    IDENTITY(1, 1),
    StartTime           TIME(6)    NOT NULL,
    EndTime             TIME(6)    NOT NULL,
    CourseScheduleId    INTEGER    NOT NULL,
    DayOfWeek           INTEGER    NOT NULL,
    CONSTRAINT          CourseDailySchedulePk   PRIMARY KEY  (DailyID)
)
;
```

```sql
-- Table: CourseGrade
CREATE TABLE vamseepr2.CourseGrade (
    GradeId         INTEGER         NOT NULL    IDENTITY(1, 1),
    Grade           CHAR(1)         NOT NULL,
    Description     NVARCHAR(30)    NOT NULL,
    CONSTRAINT      CourseGradePk   PRIMARY KEY (GradeId)
)
;

-- Table: CourseSchedule
CREATE TABLE vamseepr2.CourseSchedule (
    CourseScheduleId        INTEGER             NOT NULL    IDENTITY(1, 1),
    ClassroomId             INTEGER             NOT NULL,
    SemesterId              INTEGER             NOT NULL,
    Faculty                 INTEGER             NOT NULL,
    CourseSeats             INTEGER             NOT NULL,
    CourseCode              NVARCHAR(3)         NOT NULL,
    CourseNumber            INTEGER             NOT NULL,
    CONSTRAINT              CourseSchedulePk    PRIMARY KEY (CourseScheduleId)
)
;

-- Table: Courses
CREATE TABLE vamseepr2.Courses (
    CourseCode      NVARCHAR(3)     NOT NULL,
    CourseNumber    INTEGER         NOT NULL CHECK  (CourseNumber > 0),
    CourseTitle     NVARCHAR(50)    NOT NULL,
    Description     NVARCHAR(500)   NULL,
    Credits         INTEGER         NOT NULL,
    CONSTRAINT      CoursesPk       PRIMARY KEY     (CourseCode,CourseNumber)
)
;

-- Table: DayOfWeek
CREATE TABLE vamseepr2.DayOfWeek (
    Id          INTEGER         NOT NULL        IDENTITY(1, 1),
    Text        NVARCHAR(30)    NOT NULL,
    CONSTRAINT  DayOfWeekPk     PRIMARY KEY     (Id)
)
;

-- Table: Employees
CREATE TABLE vamseepr2.Employees (
    EmployeeId      INTEGER         NOT NULL    IDENTITY(3670000, 1),
    FirstName       NVARCHAR(50)    NOT NULL,
    LastName        NVARCHAR(50)    NOT NULL,
    NTID            NVARCHAR(25)    NOT NULL,
    SSN             NVARCHAR(max)   NOT NULL    CHECK (SSN LIKE '[1-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]'),
    AddressId       INTEGER         NOT NULL,
    YearlyPay       DECIMAL(8,2)    NOT NULL,
    JobId           INTEGER         NOT NULL,
    HealthBenefits  INTEGER         NOT NULL,
    VisionBenefits  INTEGER         NOT NULL,
    DentalBenefits  INTEGER         NOT NULL,
    IsActive        NVARCHAR(1)     NULL        DEFAULT 'Y' CHECK (IsActive IN ('Y','N')),
    CONSTRAINT      EmployeesPk     PRIMARY KEY (EmployeeId)
)
;

-- Table: Enrollment
CREATE TABLE vamseepr2.Enrollment (
    EnrollmentId        INTEGER         NOT NULL    IDENTITY(1, 1),
    StudentId           INTEGER         NOT NULL,
    GradeId             INTEGER         NOT NULL,
    CourseScheduleId    INTEGER         NOT NULL,
    GradeStatusId       INTEGER         NOT NULL,
    CONSTRAINT          EnrollmentPk    PRIMARY KEY (EnrollmentId)
)
;
```

```sql
-- Table: GradeStatus
CREATE TABLE vamseepr2.GradeStatus (
    GradeStatusId       INTEGER       NOT NULL    IDENTITY(1, 1),
    Text                NVARCHAR(15)  NOT NULL,
    CONSTRAINT          GradeStatusPk  PRIMARY KEY (GradeStatusId)
)
;

-- Table: JobInformation
CREATE TABLE vamseepr2.JobInformation (
    JobId              INTEGER           NOT NULL    IDENTITY(1, 1),
    JobTitle           NVARCHAR(50)      NOT NULL,
    Description        NVARCHAR(500)     NOT NULL,
    Requirements       NVARCHAR(1000)    NOT NULL,
    MinPay             DECIMAL(5,2)      NOT NULL,
    MaxPay             DECIMAL(7,2)      NOT NULL,
    JobPositionTypeId  INTEGER           NOT NULL,
    IsUnionJob         CHAR(1)           NOT NULL    DEFAULT 'Y' CHECK (IsUnionJob IN ('Y','N')),
    CONSTRAINT         JobInformationPk   PRIMARY KEY (JobId)
)
;

-- Table: JobPositionType
CREATE TABLE vamseepr2.JobPositionType (
    JobPositionTypeId  INTEGER           NOT NULL    IDENTITY(1, 1),
    PostionType        NVARCHAR(30)      NOT NULL,
    CONSTRAINT         JobPositionTypePk   PRIMARY KEY (JobPositionTypeId)
)
;

-- Table: Prerequisites
CREATE TABLE vamseepr2.Prerequisites (
    PrerequisiteId      INTEGER       NOT NULL    IDENTITY(1, 1),
    CourseCode          NVARCHAR(3)   NOT NULL,
    CourseNumber        INTEGER       NOT NULL,
    PrerequisiteCode    NVARCHAR(3)   NOT NULL,
    PrerequisiteNumber  INTEGER       NOT NULL,
    CONSTRAINT          PrerequisitesPk PRIMARY KEY (PrerequisiteId)
)
;

-- Table: ProgramSpecialization
CREATE TABLE vamseepr2.ProgramSpecialization (
    ProgramSpecializationId   INTEGER                NOT NULL    IDENTITY(1, 1),
    ProgramId                 INTEGER                NOT NULL,
    StudentId                 INTEGER                NOT NULL,
    IsMajor                   CHAR(1)                NOT NULL    DEFAULT 'Y' CHECK (IsMajor IN ('Y','N')),
    CONSTRAINT                ProgramSpecializationPk PRIMARY KEY (ProgramSpecializationId)
)
;

-- Table: Programs
CREATE TABLE vamseepr2.Programs (
    ProgramId    INTEGER        NOT NULL    IDENTITY(1, 1),
    ProgramName  NVARCHAR(50)   NOT NULL,
    CollegeId    INTEGER        NOT NULL,
    CONSTRAINT   ProgramsPk     PRIMARY KEY (ProgramId)
)
;

-- Table: Projector
CREATE TABLE vamseepr2.Projector (
    ProjectorId  INTEGER        NOT NULL    IDENTITY(1, 1),
    Text         NVARCHAR(30)   NOT NULL,
    CONSTRAINT   ProjectorPk    PRIMARY KEY (ProjectorId)
)
;
```

```sql
-- Table: SelectionType
CREATE TABLE vamseepr2.SelectionType (
    SelectionTypeId     INTEGER         NOT NULL    IDENTITY(1, 1),
    Selection           NVARCHAR(30)    NOT NULL,
    CONSTRAINT          SelectionTypePk PRIMARY KEY (SelectionTypeId)
)
;

-- Table: Semester
CREATE TABLE vamseepr2.Semester (
    SemesterId      INTEGER     NOT NULL    IDENTITY(1, 1),
    Year            INTEGER     NOT NULL,
    FirstDay        DATE        NOT NULL,
    LastDay         DATE        NOT NULL,
    SemesterTextId  INTEGER     NOT NULL,
    CONSTRAINT      SemesterPk  PRIMARY KEY (SemesterId)
)
;

-- Table: SemesterText
CREATE TABLE vamseepr2.SemesterText (
    SemesterTextId  INTEGER         NOT NULL    IDENTITY(1, 1),
    Text            NVARCHAR(30)    NOT NULL,
    CONSTRAINT      SemesterTextPk  PRIMARY KEY (SemesterTextId)
)
;

-- Table: StudentStatus
CREATE TABLE vamseepr2.StudentStatus (
    StudentStatusId     INTEGER         NOT NULL    IDENTITY(1, 1),
    StudentStatus       NVARCHAR(30)    NOT NULL,
    CONSTRAINT          StudentStatusPk PRIMARY KEY (StudentStatusId)
)
;

-- Table: Students
CREATE TABLE vamseepr2.Students (
    StudentId       INTEGER         NOT NULL    IDENTITY(10000, 1),
    FirstName       NVARCHAR(50)    NOT NULL,
    LastName        NVARCHAR(50)    NOT NULL,
    NTID            NVARCHAR(25)    NOT NULL,
    Password        NVARCHAR(8)     NULL,
    DateOfBirth     DATE            NOT NULL,
    SSN             NVARCHAR(max)   NULL        CHECK (SSN LIKE '[1-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]'),
    StudentStatusId INTEGER         NOT NULL,
    HomeAddress     INTEGER         NOT NULL,
    LocalAddress    INTEGER         NOT NULL,
    BillAmount      DECIMAL(8,2)    NULL        DEFAULT 0.0,
    CONSTRAINT      StudentsPk      PRIMARY KEY (StudentId)
)
;

-- foreign keys
-- Reference:  FkBuidlings (table: vamseepr2.Classroom)


ALTER TABLE vamseepr2.Classroom ADD CONSTRAINT FkBuidlings
    FOREIGN KEY     (BuildingId)
    REFERENCES      vamseepr2.Buildings (BuildingId)
;
```

```sql
-- Reference:  FkClassroom (table: vamseepr2.CourseSchedule)


ALTER TABLE vamseepr2.CourseSchedule ADD CONSTRAINT FkClassroom
    FOREIGN KEY     (ClassroomId)
    REFERENCES      vamseepr2.Classroom (ClassroomId)
;

-- Reference:  FkCollege (table: vamseepr2.Programs)


ALTER TABLE vamseepr2.Programs ADD CONSTRAINT FkCollege
    FOREIGN KEY     (CollegeId)
    REFERENCES      vamseepr2.College (CollegeId)
;

-- Reference:  FkCourseDailySchedule (table: vamseepr2.CourseDailySchedule)


ALTER TABLE vamseepr2.CourseDailySchedule ADD CONSTRAINT FkCourseDailySchedule
    FOREIGN KEY     (CourseScheduleId)
    REFERENCES      vamseepr2.CourseSchedule (CourseScheduleId)
;

-- Reference:  FkCourseScheduleCourses (table: vamseepr2.CourseSchedule)


ALTER TABLE vamseepr2.CourseSchedule ADD CONSTRAINT FkCourseScheduleCourses
    FOREIGN KEY     (CourseCode,CourseNumber)
    REFERENCES      vamseepr2.Courses (CourseCode,CourseNumber)
;

-- Reference:  FkDayOfWeek (table: vamseepr2.CourseDailySchedule)


ALTER TABLE vamseepr2.CourseDailySchedule ADD CONSTRAINT FkDayOfWeek
    FOREIGN KEY     (DayOfWeek)
    REFERENCES      vamseepr2.DayOfWeek (Id)
;

-- Reference:  FkDentalBenefits (table: vamseepr2.Employees)


ALTER TABLE vamseepr2.Employees ADD CONSTRAINT FkDentalBenefits
    FOREIGN KEY     (VisionBenefits)
    REFERENCES      vamseepr2.Benefits (BenefitId)
;

-- Reference:  FkEmployeesAddress (table: vamseepr2.Employees)


ALTER TABLE vamseepr2.Employees ADD CONSTRAINT FkEmployeesAddress
    FOREIGN KEY     (AddressId)
    REFERENCES      vamseepr2.Address (AddressId)
;

-- Reference:  FkEnrollment (table: vamseepr2.Enrollment)


ALTER TABLE vamseepr2.Enrollment ADD CONSTRAINT FkEnrollment
    FOREIGN KEY     (StudentId)
    REFERENCES      vamseepr2.Students (StudentId)
;

-- Reference:  FkEnrollmentCourseGrade (table: vamseepr2.Enrollment)

ALTER TABLE vamseepr2.Enrollment ADD CONSTRAINT FkEnrollmentCourseGrade
    FOREIGN KEY     (GradeId)
    REFERENCES      vamseepr2.CourseGrade (GradeId)
;
```

```sql
-- Reference:  FkEnrollmentCourseSchedule (table: vamseepr2.Enrollment)


ALTER TABLE vamseepr2.Enrollment ADD CONSTRAINT FkEnrollmentCourseSchedule
    FOREIGN KEY     (CourseScheduleId)
    REFERENCES      vamseepr2.CourseSchedule (CourseScheduleId)
;

-- Reference:  FkEnrollmentGradeStatus (table: vamseepr2.Enrollment)


ALTER TABLE vamseepr2.Enrollment ADD CONSTRAINT FkEnrollmentGradeStatus
    FOREIGN KEY     (GradeStatusId)
    REFERENCES      vamseepr2.GradeStatus (GradeStatusId)
;

-- Reference:  FkFaculty (table: vamseepr2.CourseSchedule)


ALTER TABLE vamseepr2.CourseSchedule ADD CONSTRAINT FkFaculty
    FOREIGN KEY     (Faculty)
    REFERENCES      vamseepr2.Employees (EmployeeId)
;

-- Reference:  FkHealthBenefits (table: vamseepr2.Employees)


ALTER TABLE vamseepr2.Employees ADD CONSTRAINT FkHealthBenefits
    FOREIGN KEY     (HealthBenefits)
    REFERENCES      vamseepr2.Benefits (BenefitId)
;

-- Reference:  FkJobInformation (table: vamseepr2.Employees)


ALTER TABLE vamseepr2.Employees ADD CONSTRAINT FkJobInformation
    FOREIGN KEY     (JobId)
    REFERENCES      vamseepr2.JobInformation (JobId)
;

-- Reference:  FkJobPositionType (table: vamseepr2.JobInformation)


ALTER TABLE vamseepr2.JobInformation ADD CONSTRAINT FkJobPositionType
    FOREIGN KEY     (JobPositionTypeId)
    REFERENCES      vamseepr2.JobPositionType (JobPositionTypeId)
;

-- Reference:  FkParentCourses (table: vamseepr2.Prerequisites)


ALTER TABLE vamseepr2.Prerequisites ADD CONSTRAINT FkParentCourses
    FOREIGN KEY     (CourseCode,CourseNumber)
    REFERENCES      vamseepr2.Courses (CourseCode,CourseNumber)
;

-- Reference:  FkPrerequisitesCourses (table: vamseepr2.Prerequisites)


ALTER TABLE vamseepr2.Prerequisites ADD CONSTRAINT FkPrerequisitesCourses
    FOREIGN KEY     (PrerequisiteCode,PrerequisiteNumber)
    REFERENCES      vamseepr2.Courses (CourseCode,CourseNumber)
;
```

```
-- Reference:  FkProgramSpecilization (table: vamseepr2.ProgramSpecialization)


ALTER TABLE vamseepr2.ProgramSpecialization ADD CONSTRAINT FkProgramSpecilization
    FOREIGN KEY      (StudentId)
    REFERENCES       vamseepr2.Students (StudentId)
;

-- Reference:  FkPrograms (table: vamseepr2.ProgramSpecialization)


ALTER TABLE vamseepr2.ProgramSpecialization ADD CONSTRAINT FkPrograms
    FOREIGN KEY      (ProgramId)
    REFERENCES       vamseepr2.Programs (ProgramId)
;

-- Reference:  FkProjector (table: vamseepr2.Classroom)


ALTER TABLE vamseepr2.Classroom ADD CONSTRAINT FkProjector
    FOREIGN KEY      (ProjectorId)
    REFERENCES       vamseepr2.Projector (ProjectorId)
;

-- Reference:  FkSectionSemester (table: vamseepr2.CourseSchedule)


ALTER TABLE vamseepr2.CourseSchedule ADD CONSTRAINT FkSectionSemester
    FOREIGN KEY      (SemesterId)
    REFERENCES       vamseepr2.Semester (SemesterId)
;

-- Reference:  FkSelectionType (table: vamseepr2.Benefits)


ALTER TABLE vamseepr2.Benefits ADD CONSTRAINT FkSelectionType
    FOREIGN KEY      (SelectionType)
    REFERENCES       vamseepr2.SelectionType (SelectionTypeId)
;

-- Reference:  FkSemesterText (table: vamseepr2.Semester)


ALTER TABLE vamseepr2.Semester ADD CONSTRAINT FkSemesterText
    FOREIGN KEY      (SemesterTextId)
    REFERENCES       vamseepr2.SemesterText (SemesterTextId)
;

-- Reference:  FkStudentStatus (table: vamseepr2.Students)


ALTER TABLE vamseepr2.Students ADD CONSTRAINT FkStudentStatus
    FOREIGN KEY      (StudentStatusId)
    REFERENCES       vamseepr2.StudentStatus (StudentStatusId)
;

-- Reference:  FkStudentsHomeAddress (table: vamseepr2.Students)


ALTER TABLE vamseepr2.Students ADD CONSTRAINT FkStudentsHomeAddress
    FOREIGN KEY      (HomeAddress)
    REFERENCES       vamseepr2.Address (AddressId)
;

-- Reference:  FkStudentsLocalAddress (table: vamseepr2.Students)


ALTER TABLE vamseepr2.Students ADD CONSTRAINT FkStudentsLocalAddress
    FOREIGN KEY      (LocalAddress)
    REFERENCES       vamseepr2.Address (AddressId)
```

```
;

-- Reference:  FkVisionBenefits (table: vamseepr2.Employees)


ALTER TABLE vamseepr2.Employees ADD CONSTRAINT FkVisionBenefits
    FOREIGN KEY     (DentalBenefits)
    REFERENCES      vamseepr2.Benefits (BenefitId)
;

-- End of file.
```