DROP table if exists NS\_WorksInsurance cascade;

DROP table if exists NS\_Themes cascade;

DROP table if exists NS\_SponsorExhibitions cascade;

DROP table if exists NS\_WorkTransaction cascade;

DROP table if exists NS\_ExhibitionLocations cascade;

DROP table if exists NS\_ExhibitionWorks cascade;

DROP table if exists NS\_WorkOwners cascade;

DROP table if exists NS\_WorksMedium cascade;

DROP table if exists NS\_Works cascade;

DROP table if exists NS\_Owners cascade;

DROP table if exists NS\_Locations cascade;

DROP table if exists NS\_Exhibitions cascade;

DROP table if exists NS\_Doors cascade;

DROP table if exists NS\_WorkLocations cascade;

--Need to create a simple data attribute for phone numbers. If you don’t have them, don’t worry!

--CREATE DOMAIN NS\_PhoneNumber as varchar(13);

--ALTER TABLE ns\_owners add column ownphonenumber NS\_PhoneNumber;

CREATE TABLE NS\_Owners (

ownName NS\_OwnerName NOT NULL,

ownEmail NS\_OwnerEmail,

ownCity NS\_OwnerCity,

ownState NS\_OwnerState,

ownStreetName NS\_OwnerStreetName,

ownBuildingNumber NS\_OwnerBuildingNumber,

ownisPartnerMuseum NS\_IsPartnerMuseum NOT NULL,

PRIMARY KEY(ownName)

);

CREATE TABLE NS\_Works (

worWorkCharID NS\_WorkCharID NOT NULL,

worWorkNumID NS\_WorkNumID NOT NULL,

worDatabaseEntryMuseum NS\_OwnerName NOT NULL,

worWorkName NS\_WorkName,

worWorkPhysicalProperty NS\_WorkPhysicalProperty,

worWorkClassification NS\_WorkClassification,

worWorkCreator NS\_WorkCreator,

worWorkCreationDate NS\_WorkCreationDate,

worWorkBorrowable NS\_WorkBorrowable NOT NULL,

worWorkGeographicRegionOfOrigin NS\_GeographicRegionOfOrigin,

worWorkCountryOfOrigin NS\_CountryOfOrigin,

worWorkFieldOfScience NS\_FieldOfScience,

worWorkDescription NS\_WorkDescription, -- mk

PRIMARY KEY(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum),

FOREIGN KEY(worDatabaseEntryMuseum) REFERENCES NS\_Owners(ownName) ON UPDATE CASCADE

);

CREATE TABLE NS\_WorksMedium (

wmeWorkCharID NS\_WorkCharID NOT NULL,

wmeWorkNumID NS\_WorkNumID NOT NULL,

wmeDatabaseEntryMuseum NS\_OwnerName NOT NULL,

wmeWorkMedium NS\_WorkMedium NOT NULL,

PRIMARY KEY(wmeWorkCharID, wmeWorkNumID, wmeDatabaseEntryMuseum, wmeWorkMedium),

FOREIGN KEY(wmeWorkCharID, wmeWorkNumID, wmeDatabaseEntryMuseum) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum) ON UPDATE CASCADE

);

--Call locBuildingName locOwnerName (building name too vague)

CREATE TABLE NS\_Locations (

locLocationName NS\_LocationName, --was RoomName but could be towns

locBuildingName NS\_OwnerName, --will have to be owner name and the name building name should change

locIsTravellingExhibitionLocation NS\_IsTravellingExhibitionLocation,

locLocationSuggestedCapacityMin NS\_LocationSuggestedCapacity,

locLocationSuggestedCapacityMax NS\_LocationSuggestedCapacity,

locLocationMeasurementWidth NS\_LocationMeasurement,

locLocationMeasurementLength NS\_LocationMeasurement,

FOREIGN KEY(locBuildingName) REFERENCES NS\_Owners(ownName),

PRIMARY KEY(locLocationName, locBuildingName),

CHECK(locLocationSuggestedCapacityMin < locLocationSuggestedCapacityMax)

);

CREATE TABLE NS\_WorkLocations (

wolWorkCharID NS\_WorkCharID NOT NULL,

wolWorkNumID NS\_WorkNumID NOT NULL,

wolWorkDatabaseEntryMuseum NS\_OwnerName NOT NULL,

wolLocationName NS\_LocationName NOT NULL,

wolMuseumName NS\_OwnerName NOT NULL,

wolWorkTimeArrival NS\_WorkTime NOT NULL,

wolWorkTimeDeparture NS\_WorkTime,

Foreign Key (wolWorkCharID, wolWorkNumID, wolWorkDatabaseEntryMuseum) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum),

Foreign Key (wolLocationName, wolMuseumName) REFERENCES NS\_Locations(locLocationName, locBuildingName),

PRIMARY KEY(wolWorkCharID, wolWorkNumID, wolWorkDatabaseEntryMuseum, wolLocationName, wolMuseumName,wolWorkTimeArrival),

CHECK ((wolWorkTimeArrival<=wolWorkTimeDeparture) OR (wolWorkTimeDeparture is NULL))

);

CREATE TABLE NS\_Doors (

dorMuseumNameOrigin NS\_OwnerName NOT NULL,

dorLocationNameOrigin NS\_LocationName NOT NULL,

dorMuseumNameDestination NS\_OwnerName NOT NULL,

dorLocationNameDestination NS\_LocationName NOT NULL,

FOREIGN KEY(dorMuseumNameOrigin, dorLocationNameOrigin) REFERENCES NS\_Locations(locBuildingName, locLocationName),

FOREIGN KEY(dorMuseumNameDestination, dorLocationNameDestination) REFERENCES NS\_Locations(locBuildingName, locLocationName),

PRIMARY KEY(dorMuseumNameOrigin, dorLocationNameOrigin, dorMuseumNameDestination,dorLocationNameDestination)

);

**--Missing Descriptions!!!**

CREATE TABLE NS\_Exhibitions (

exhExhibitName NS\_ExhibitName NOT NULL,

exhMuseumName NS\_OwnerName NOT NULL,

exhExhibitDateStart NS\_ExhibitDate NOT NULL,

exhExhibitDateEnd NS\_ExhibitDate NOT NULL,

exhExhibitDescription NS\_ExhibitDescription, -- Added description

exhSecurityPersonName NS\_SecurityName,

exhIsTravellingExhibition NS\_IsTravellingExhibition,

PRIMARY KEY(exhExhibitName, exhMuseumName, exhExhibitDateStart),

FOREIGN KEY(exhMuseumName) REFERENCES NS\_Owners(OwnName),

CHECK( exhExhibitDateStart<exhExhibitDateEnd)

);

CREATE TABLE NS\_ExhibitionLocations (

eloExhibitName NS\_ExhibitName NOT NULL,

eloExhibitDateStart NS\_ExhibitDate NOT NULL,

eloLocationName NS\_LocationName NOT NULL,

eloMuseumOfLocation NS\_OwnerName NOT NULL,

eloMuseumOfExhibition NS\_OwnerName NOT NULL,

eloLocationDateStart NS\_LocationDate NOT NULL,

eloLocationDateEnd NS\_LocationDate,

PRIMARY KEY(eloExhibitName, eloExhibitDateStart, eloLocationName, eloMuseumOfLocation, eloLocationDateStart),

FOREIGN KEY(eloExhibitname, eloExhibitDateStart,eloMuseumOfExhibition) REFERENCES NS\_Exhibitions(exhExhibitName, exhExhibitDateStart, exhMuseumName),

FOREIGN KEY(eloLocationName, eloMuseumOfLocation) REFERENCES NS\_Locations(locLocationName, locBuildingName),

CHECK(((eloLocationDateEnd is NULL) OR (eloLocationDateStart<eloLocationDateEnd))

AND (eloExhibitDateStart<=eloLocationDateStart))

);

CREATE TABLE NS\_ExhibitionWorks (

exwExhibitName NS\_ExhibitName NOT NULL,

exwExhibitDateStart NS\_ExhibitDate NOT NULL,

exwMuseumOfExhibition NS\_OwnerName NOT NULL,

exwWorkCharID NS\_WorkCharID NOT NULL,

exwWorkNumID NS\_WorkNumID NOT NULL,

exwWorkDataBaseEntryMuseum NS\_OwnerName NOT NULL,

exwDateWorkAdded NS\_DateWorkAdded NOT NULL,

exwDateWorkRemoved NS\_DateWorkRemoved,

PRIMARY KEY(exwExhibitname, exwExhibitDateStart, exwMuseumOfExhibition, exwWorkCharID, exwWorkNumID, exwWorkDataBaseEntryMuseum, exwDateWorkAdded),

FOREIGN KEY(exwExhibitName, exwExhibitDateStart,exwMuseumOfExhibition) REFERENCES NS\_Exhibitions(exhExhibitName,exhExhibitDateStart, exhMuseumName),

FOREIGN KEY(exwWorkCharID, exwWorkNumID, exwWorkDatabaseEntryMuseum) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum),

CHECK((exwDateWorkAdded<exwDateWorkRemoved) OR (exwDateWorkRemoved is NULL))

);

--Change EntryLocation to EntryMuseum?

CREATE TABLE NS\_WorkOwners (

wonWorkCharID NS\_WorkCharID NOT NULL,

wonWorkNumID NS\_WorkNumID NOT NULL,

wonWorkDatabaseEntryLocation NS\_OwnerName NOT NULL,

wonOwnerName NS\_OwnerName NOT NULL,

wonOwnershipDateStart NS\_OwnershipDate NOT NULL,

wonOwnershipDateEnd NS\_OwnershipDate,

PRIMARY KEY(wonWorkCharID, wonWorkNumId, wonworkDatabaseEntryLocation, wonOwnerName, wonOwnershipDateStart),

FOREIGN KEY(wonWorkCharID, wonWorkNumID, wonWorkDatabaseEntryLocation) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum),

FOREIGN KEY(wonOwnerName) REFERENCES NS\_Owners(ownName),

CHECK((wonOwnershipDateStart<wonOwnershipDateEnd) OR (wonOwnershipDateEnd is NULL))

);

CREATE TABLE NS\_WorkTransaction (

wtrWorkCharID NS\_WorkCharID NOT NULL,

wtrWorkNumID NS\_WorkNumID NOT NULL,

wtrWorkDatabaseEntryMuseum NS\_OwnerName NOT NULL,

wtrMuseumInvolved NS\_OwnerName NOT NULL,

wtrClient NS\_OwnerName,

wtrTransactiontype NS\_TransactionType NOT NULL,

wtrTransactionTime NS\_TransactionTime NOT NULL,

PRIMARY KEY(wtrWorkCharID, wtrWorkNumID, wtrWorkDatabaseEntryMuseum, wtrMuseumInvolved, wtrTransactionType, wtrTransactionTime),

FOREIGN KEY(wtrWorkCharID, wtrWorkNumID, wtrWorkDatabaseEntryMuseum) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum),

FOREIGN KEY(wtrWorkDatabaseEntryMuseum) REFERENCES NS\_Owners(ownName),

FOREIGN KEY(wtrClient) REFERENCES NS\_Owners(ownName)

);

CREATE TABLE NS\_SponsorExhibitions (

speSponsorName NS\_SponsorName NOT NULL,

speExhibitName NS\_ExhibitName NOT NULL,

speMuseumName NS\_OwnerName NOT NULL,

speExhibitDateStart NS\_ExhibitDate NOT NULL,

speSponsorAmount NS\_SponsorAmount,

PRIMARY KEY(speSponsorName, speExhibitName, speMuseumName, speExhibitDateStart),

FOREIGN KEY(speExhibitName, speMuseumName, speExhibitDateStart) REFERENCES NS\_Exhibitions(exhExhibitName, exhMuseumName, exhExhibitDateStart)

);

--Change EntryLocation to EntryMuseum

CREATE TABLE NS\_Themes (

wthWorkCharID NS\_WorkCharID NOT NULL,

wthWorkNumID NS\_WorkNumID NOT NULL,

wthWorkDatabaseEntryLocation NS\_OwnerName NOT NULL,

wthTheme NS\_Theme NOT NULL,

PRIMARY KEY(wthWorkCharID, wthWorkNumID, wthWorkDatabaseEntryLocation, wthTheme),

FOREIGN KEY(wthWorkCharID, wthWorkNumID, wthWorkDatabaseEntryLocation) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum)

);

--Change EntryLocation to EntryMuseum?

CREATE TABLE NS\_WorksInsurance (

winWorkCharID NS\_WorkCharID NOT NULL,

winWorkNumID NS\_WorkNumID NOT NULL,

winWorkDatabaseEntryLocation NS\_OwnerName NOT NULL,

winWorkInsureValue NS\_WorkInsureValue NOT NULL,

winInsureDateStart NS\_InsureDate NOT NULL,

winInsureDateEnd NS\_InsureDate,

PRIMARY KEY(winWorkCharID, winWorkNumID, winWorkDatabaseEntryLocation, winInsureDateStart),

FOREIGN KEY(winWorkCharID, winWorkNumID, winWorkDatabaseEntryLocation) REFERENCES NS\_Works(worWorkCharID, worWorkNumID, worDatabaseEntryMuseum),

CHECK((winInsureDateStart< winInsureDateEnd) OR (winInsureDateEnd is NULL))

);

--We can add this at merge time

--RoomCondition

--Table