Database

CSE308: Software Engineering
Section 1
Professor Scott Stoller
March 24, 2009

Jonathan Farrell

Claude Castille

Timothy Quinto

Michael Wu (Hung Ngo)

For all documentation and progress, please visit us at:
WWW.CORPUSVILE.COM/CRISYS

This document contains the database information for the CRISYS Electronic Health Records system. This Document was completed March 2009 at Stony Brook University

Contents

Persistence

We will be using a MySQL database on our web server for all database transactions. The following is our ER Diagram for conceptual design of our database and a copy of our SQL CREATE script which is run on the server to configure the database.

ER Diagram:

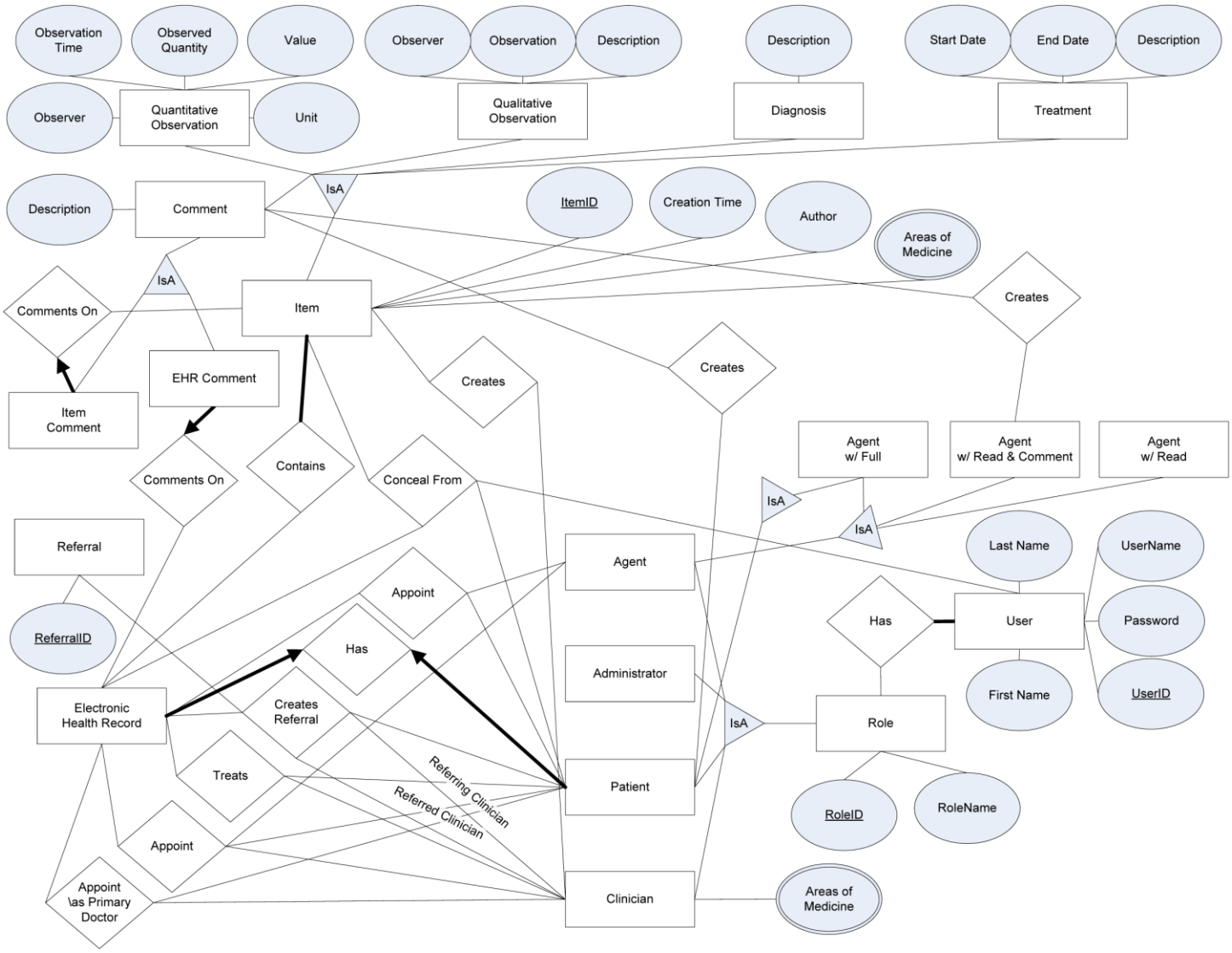

Table Descriptions

Table Descriptions

Table:	Administrator		--table identifies which users are administrators of the system.
Attributes:	ID	INT	--ID Number of the user, should be in the User table.

Table:	Appointed_Agents		--only table that identifies users as agents. Holds information on the user, the associated health record (the idea is that agents are associated with the record not the patient), and the role they have over that record
Attributes:	agent	INT	--UserID of the agent
	electronicHealthRecord	INT	--EHR the agent is sponsoring
	role	INT	--role of the agent over the record (should be one of the agent role IDs)

Table:	Clinician		--table identifies users as Clinicians
Attributes:	ID	INT	--ID number of the user, should be in the User table.

Table:	Clinician_Medical_Areas		--keeps a record of the medical areas of each clinician
Attributes:	Clinician	INT	--userId of the clinician
	MedicalArea	INT	--medical area ID of the clinician's medical area

Table:	Comment_Item		--represents a comment item in the HER
Attributes:	ID	INT	--itemID should map back to an ID in the item table
	description	VARCHAR(255)	--comment contents
	commentedItem	INT	--commented item, can be null, if it is null then it's a comment on the EHR itself

Table:	Concealed_Item		--item being blocked and who it's being blocked from
Attributes:	item	INT	--item ID of the item being concealed
	blockedUser	INT	--userID of the user this item is being blocked from

Table:	Diagnosis_Item		--table that represents diagnosis item
Attributes:	ID	INT	--item ID of the diagnosis item, maps to ID in item table
	description	VARCHAR(255)	--diagnosis description

Table:	Electrionic_Health_Record_Items		--mapping of items to the heath record they belong to
Attributes:	EHRId	INT	--ID of the EHR, maps to an ID in the EHR table
	item	INT	--item associated with this EHR maps to an item in the item table

Table:	Electronic_Health_Record		--table that identifies an electronic health record object (currently only holds an ID for the object, but additional information can be added later)
Attributes:	id	INT	--ID of the EHR

Table:	Item		--table that represents item objects
Attributes:	ID	INT	--id of the item
	ItemType	INT	--type of the item, maps to the item types table (seems redundant, since membership in the table implies item type)
	Author	INT	--user id of the author of the item
	CreationTime	DATETIME	--date and time the item was created

Table:	Item_Medical_Areas		-- keeps a record of the medical areas of each item
Attributes:	ItemID	INT	--id of the item
	MedicalArea	INT	--medical area associated with the item

Table:	Item_Types		--table of item type descriptions
Attributes:	Id	INT	--id of the item type
	description	VARCHAR(255)	--description of the item type

Table:	Log		--table to store system logs
Attributes:	ID	INT	--id of the log
	electronicHealthRecord	INT	--id of the associated EHR (can be null)
	editingUser	INT	--user who initiated the action
	dateOfAction	DATETIME	--date and time of the action
	actionDescription	VARCHAR(255)	--description of the action

Table:	Medical_Areas		--list of medical areas
Attributes:	Id		--id identifying a medical area
	Description		--description of a medical area

Table:	Observed_Quantitity_Type		--list of observed quantities (in case we go for a drop down instead of a text field for the observed quantity type)
Attributes:	Id		--id of the observed quantity type
	Description		--description of the observed quantity type

Table:	Patient		--table identifies users as patients
Attributes:	Id	INT	--id of the patient, maps to a record in the user table
	electronicHealthRecord	INT	--EHR associated with this patient

Table:	Primary_Doctor		--table that associates primary doctors with EHRs (note that doctors aren't assigned to the patients, they're assigned to the record)
Attributes:	electronicHealthRecord	INT	--id of the EHR the clinician is the primary physician for
	primaryDoctor	INT	--primary physician for this HER

Table:	Qualitative_Observation_Item		--table that stores qualitative observation items
Attributes:	Id	INT	--id of the observation item, maps to item id in item
	Observer	INT	--id of user that performed the observation, maps to user table
	ObservationTime	DATETIME	--date and time of the observation
	Description	VARCHAR(255)	--description of the observation

Table:	Quantitative_Observation_Item		--table that stores quantitative observation items
Attributes:	Id	INT	--id of the observation item, maps to item id in item
	Observer	INT	--id of user that performed the observation, maps to user table
	observedQuantity	INT	--id of the observed quantity type, maps to the observed quantity type table
	Value	INT	--value of the observation
	unit	VARCHAR(255)	--unit of measurement

Table:	Referral		--table that stores the referrals in the system
Attributes:	Id	INT	--id of the referral
	electronicHealthRecord	INT	--id of the EHR, maps to the EHR table
	referringClinician	INT	--id of the clinician making the referral, maps to user table
	referredClinician	INT	--if of the clinician being referred, maps to user table
	Author	INT	--id of the author of the item, maps to user table
	creationTime	DATETIME	--datetime of the creation of the referral
	Cancelled	TINYINT	--0 if this referral hasn't been cancelled, 1 otherwise
	CancelledBy	INT	--userid of the user that cancelled the referral, maps to user table

Table:	Role		--Access Control Matrix facilitating Role Based Access Control for the system. Has descriptions of the roles and every ability a user can perform.
Attributes:	id	INT	--id of the role
	RoleName	VARCHAR(45)	--name of the role
	RoleDisplayName	VARCHAR(45)	--display name of the role
	SearchItemsInEHR	TINYINT	--1 if the role has the ability, 0 otherwise
	BrowseItemsInEHR	TINYINT	--1 if the role has the ability, 0 otherwise
	SearchAgents	TINYINT	--1 if the role has the ability, 0 otherwise
	BrowseAgents	TINYINT	--1 if the role has the ability, 0 otherwise
	SearchSponsoredPatients	TINYINT	--1 if the role has the ability, 0 otherwise
	BrowseSponsoredPatients	TINYINT	--1 if the role has the ability, 0 otherwise
	SearchPatients	TINYINT	--1 if the role has the ability, 0 otherwise
	BrowsePatients	TINYINT	--1 if the role has the ability, 0 otherwise
	AppointAgent	TINYINT	--1 if the role has the ability, 0 otherwise
	RevokeAgent	TINYINT	--1 if the role has the ability, 0 otherwise
	RevokeCreatedAgent	TINYINT	--1 if the role has the ability, 0 otherwise
	EditAgentPrivileges	TINYINT	--1 if the role has the ability, 0 otherwise
	AddQuantitativeObservationItem	TINYINT	--1 if the role has the ability, 0 otherwise
	AddQualitativeObservationItem	TINYINT	--1 if the role has the ability, 0 otherwise
	AddCommentItemToItem	TINYINT	--1 if the role has the ability, 0 otherwise
	AddCommentItem	TINYINT	--1 if the role has the ability, 0 otherwise
	AddTreatmentItem	TINYINT	--1 if the role has the ability, 0 otherwise
	AddDiagnosisItem	TINYINT	--1 if the role has the ability, 0 otherwise
	ConcealItem	TINYINT	--1 if the role has the ability, 0 otherwise
	CancelItemConcealment	TINYINT	--1 if the role has the ability, 0 otherwise
	SetDotctorAsPrimary	TINYINT	--1 if the role has the ability, 0 otherwise
	RemoveDoctorAsPrimary	TINYINT	--1 if the role has the ability, 0 otherwise
	GrantConsentToTreatment	TINYINT	--1 if the role has the ability, 0 otherwise
	WithdrawConsentToTreatment	TINYINT	--1 if the role has the ability, 0 otherwise
	AddUserToSystem	TINYINT	--1 if the role has the ability, 0 otherwise
	AddDoctorAsPrimary	TINYINT	--1 if the role has the ability, 0 otherwise
	RevokeDoctorAsPrimary	TINYINT	--1 if the role has the ability, 0 otherwise
	CreateReferral	TINYINT	--1 if the role has the ability, 0 otherwise
	CancelReferral	TINYINT	--1 if the role has the ability, 0 otherwise
	CancelCreatedReferral	TINYINT	--1 if the role has the ability, 0 otherwise
	CreateMedicalArea	TINYINT	--1 if the role has the ability, 0 otherwise
	CreateObservedQuantityType	TINYINT	--1 if the role has the ability, 0 otherwise

Table:	Treating_Clinicians		--table to keep track of the clinician in charge of treating each EHR
Attributes:	electronicHealthRecord	INT	--id of the EHR, maps to the EHR table
	Clinician	INT	--id of the clinician, maps to the user table

Table:	Treatment_Item		--table to keep track of the treatment items
Attributes:	id	INT	--id of the treatment item, maps to item table
	startDate	DATETIME	--datetime of the treatment start
	endDate	DATETIME	--datetime of the treatment end
	description	VARCHAR(255)	--description of the item

Table:	User		--table that identifies users of the system
Attributes:	Id	INT	--id that uniquely identifies a user of the system
	firstName	VARCHAR(45)	--first name of a user
	lastName	VARCHAR(45)	--last name of a user
	Username	VARCHAR(255)	--unique username of a user
	Password	VARCHAR(255)	--hashed and password of a user
	Active	TINYINT	--whether the user is active in the system. 1 for active, 0 otherwise.

SQL CREATE Script:

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `electronic_health_record_system` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;
USE `electronic_health_record_system`;

-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`User`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`User` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  `FirstName` VARCHAR(45) NOT NULL ,
  `LastName` VARCHAR(45) NOT NULL ,
  `Username` VARCHAR(255) NOT NULL ,
  `Password` VARCHAR(255) NOT NULL ,
  `Active` TINYINT NOT NULL DEFAULT 1 ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB;

CREATE UNIQUE INDEX `unique_username` USING BTREE ON `electronic_health_record_system`.`User` (`Username` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Role`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Role` (
  `ID` INT NOT NULL ,
  `RoleName` VARCHAR(45) NOT NULL ,
  `RoleDisplayName` VARCHAR(45) NOT NULL ,
  `SearchItemsInEHR` TINYINT NOT NULL ,
  `BrowseItemsInEHR` TINYINT NOT NULL ,
  `SearchAgents` TINYINT NOT NULL ,
  `BrowseAgents` TINYINT NOT NULL ,
  `SearchSponsoredPatients` TINYINT NOT NULL ,
  `BrowseSponsoredPatients` TINYINT NOT NULL ,
  `SearchPatients` TINYINT NOT NULL ,
  `BrowsePatients` TINYINT NOT NULL ,
  `AppointAgent` TINYINT NOT NULL ,
  `RevokeAgent` TINYINT NOT NULL ,
  `RevokeCreatedAgent` TINYINT NOT NULL ,
  `EditAgentPrivileges` TINYINT NOT NULL ,
  `AddQuantitativeObservationItem` TINYINT NOT NULL ,
  `AddQualitativeObservationItem` TINYINT NOT NULL ,
  `AddCommentItemToItem` TINYINT NOT NULL ,
  `AddCommentItem` TINYINT NOT NULL ,
  `AddTreatmentItem` TINYINT NOT NULL ,
  `AddDiagnosisItem` TINYINT NOT NULL ,
  `ConcealItem` TINYINT NOT NULL ,
  `CancelItemConcealment` TINYINT NOT NULL ,
  `SetDotctorAsPrimary` TINYINT NOT NULL ,
  `RemoveDoctorAsPrimary` TINYINT NOT NULL ,
  `GrantConsentToTreatment` TINYINT NOT NULL ,
  `WithdrawConsentToTreatment` TINYINT NOT NULL ,
  `AddUserToSystem` TINYINT NOT NULL ,
  `AddDoctorAsPrimary` TINYINT NOT NULL ,
  `RevokeDoctorAsPrimary` TINYINT NOT NULL ,
  `CreateReferral` TINYINT NOT NULL ,
  `CancelReferral` TINYINT NOT NULL ,
  `CancelCreatedReferral` TINYINT NOT NULL ,
```

```
  `CreateMedicalArea` TINYINT NOT NULL ,
  `CreateObservedQuantityType` TINYINT NOT NULL ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Medical_Areas`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Medical_Areas` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  `Description` VARCHAR(255) NOT NULL ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Item_Types`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Item_Types` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  `Description` VARCHAR(255) NULL ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Item` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  `ItemType` INT NULL ,
  `Author` INT NULL ,
  `CreationTime` DATETIME NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Item_itemTypeIDToItemType`
    FOREIGN KEY (`ItemType` )
    REFERENCES `electronic_health_record_system`.`Item_Types` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Item_authorToUserID`
    FOREIGN KEY (`Author` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Item_itemTypeIDToItemType` ON `electronic_health_record_system`.`Item` (`ItemType` ASC) ;

CREATE INDEX `Item_authorToUserID` ON `electronic_health_record_system`.`Item` (`Author` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Item_Medical_Areas`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Item_Medical_Areas` (
  `ItemID` INT NOT NULL ,
  `MedicalArea` INT NOT NULL ,
  PRIMARY KEY (`ItemID`, `MedicalArea`) ,
  CONSTRAINT `Item_Medical_Areas_itemIDToItemID`
    FOREIGN KEY (`ItemID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Item_Medical_Areas_medicalAreaToMedicalAreaID`
    FOREIGN KEY (`MedicalArea` )
    REFERENCES `electronic_health_record_system`.`Medical_Areas` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Item_Medical_Areas_itemIDToItemID` ON `electronic_health_record_system`.`Item_Medical_Areas` (`ItemID` ASC) ;

CREATE INDEX `Item_Medical_Areas_medicalAreaToMedicalAreaID` ON `electronic_health_record_system`.`Item_Medical_Areas` (`MedicalArea`
ASC) ;
```

```
-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Observed_Quantitity_Type`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Observed_Quantitity_Type` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  `Description` VARCHAR(255) NOT NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Observed_Quantitity_Type_idToItemID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Observed_Quantitity_Type_idToItemID` ON `electronic_health_record_system`.`Observed_Quantitity_Type` (`ID` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Quantitative_Observation_Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Quantitative_Observation_Item` (
  `ID` INT NOT NULL ,
  `Observer` INT NULL ,
  `ObservedQuantity` INT NULL ,
  `Value` INT NULL ,
  `Unit` VARCHAR(255) NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Quantitative_Obser_Item_idToItemID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Quantitative_Obser_Item_observerToUserID`
    FOREIGN KEY (`Observer` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Quantitative_Obser_Item_observedQuantityToObservedQuantityTypeID`
    FOREIGN KEY (`ObservedQuantity` )
    REFERENCES `electronic_health_record_system`.`Observed_Quantitity_Type` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Quantitative_Obser_Item_idToItemID` ON `electronic_health_record_system`.`Quantitative_Observation_Item` (`ID` ASC) ;

CREATE INDEX `Quantitative_Obser_Item_observerToUserID` ON `electronic_health_record_system`.`Quantitative_Observation_Item` (`Observer`
ASC) ;

CREATE INDEX `Quantitative_Obser_Item_observedQuantityToObservedQuantityTypeID` ON
`electronic_health_record_system`.`Quantitative_Observation_Item` (`ObservedQuantity` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Qualitative_Observation_Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Qualitative_Observation_Item` (
  `ID` INT NOT NULL ,
  `Observer` INT NULL ,
  `ObservationTime` DATETIME NULL ,
  `Description` VARCHAR(255) NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Qualitative_Obser_Item_idToItemID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Qualitative_Obser_Item_observerToUserID`
    FOREIGN KEY (`Observer` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
CREATE INDEX `Qualitative_Obser_Item_idToItemID` ON `electronic_health_record_system`.`Qualitative_Observation_Item` (`ID` ASC) ;

CREATE INDEX `Qualitative_Obser_Item_observerToUserID` ON `electronic_health_record_system`.`Qualitative_Observation_Item` (`Observer` ASC)
;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Diagnosis_Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Diagnosis_Item` (
  `ID` INT NOT NULL ,
  `description` VARCHAR(255) NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Diagnosis_Item_idToItemID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Diagnosis_Item_idToItemID` ON `electronic_health_record_system`.`Diagnosis_Item` (`ID` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Treatment_Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Treatment_Item` (
  `ID` INT NOT NULL ,
  `startDate` DATETIME NULL ,
  `endDate` DATETIME NULL ,
  `description` VARCHAR(255) NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Treatment_Item_idToItemID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Treatment_Item_idToItemID` ON `electronic_health_record_system`.`Treatment_Item` (`ID` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Comment_Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Comment_Item` (
  `ID` INT NOT NULL ,
  `description` VARCHAR(255) NULL ,
  `commentedItem` INT NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Comment_Item_idToItemID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Comment_Item_commentedItemToItemID`
    FOREIGN KEY (`commentedItem` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Comment_Item_idToItemID` ON `electronic_health_record_system`.`Comment_Item` (`ID` ASC) ;

CREATE INDEX `Comment_Item_commentedItemToItemID` ON `electronic_health_record_system`.`Comment_Item` (`commentedItem` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Electronic_Health_Record`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Electronic_Health_Record` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Patient`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Patient` (
  `ID` INT NOT NULL ,
  `electronicHealthRecord` INT NOT NULL ,
  PRIMARY KEY (`electronicHealthRecord`, `ID`) ,
  CONSTRAINT `Patient_idToUserID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Patient_EHRToEHRID`
    FOREIGN KEY (`electronicHealthRecord` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE UNIQUE INDEX `patient_ID` USING HASH ON `electronic_health_record_system`.`Patient` (`ID` ASC) ;

CREATE UNIQUE INDEX `EHR_ID` USING HASH ON `electronic_health_record_system`.`Patient` (`electronicHealthRecord` ASC) ;

CREATE INDEX `Patient_idToUserID` ON `electronic_health_record_system`.`Patient` (`ID` ASC) ;

CREATE INDEX `Patient_EHRToEHRID` ON `electronic_health_record_system`.`Patient` (`electronicHealthRecord` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Electrionic_Health_Record_Items`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Electrionic_Health_Record_Items` (
  `EHRID` INT NOT NULL ,
  `Item` INT NOT NULL ,
  PRIMARY KEY (`EHRID`, `Item`) ,
  CONSTRAINT `Electrionic_Health_Record_Items_EHRIDToEHRID`
    FOREIGN KEY (`EHRID` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Electrionic_Health_Record_Items_itemToItemID`
    FOREIGN KEY (`Item` )
    REFERENCES `electronic_health_record_system`.`Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE UNIQUE INDEX `uniqueItemID` ON `electronic_health_record_system`.`Electrionic_Health_Record_Items` (`Item` ASC) ;

CREATE INDEX `Electrionic_Health_Record_Items_EHRIDToEHRID` ON `electronic_health_record_system`.`Electrionic_Health_Record_Items`
(`EHRID` ASC) ;

CREATE INDEX `Electrionic_Health_Record_Items_itemToItemID` ON `electronic_health_record_system`.`Electrionic_Health_Record_Items` (`Item`
ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Administrator`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Administrator` (
  `ID` INT NOT NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `idToUserID`
    FOREIGN KEY (`ID` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `idToUserID` ON `electronic_health_record_system`.`Administrator` (`ID` ASC) ;


-- -----------------------------------------------------
```

```
-- Table `electronic_health_record_system`.`Referral`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Referral` (
  `ID` INT NOT NULL AUTO_INCREMENT ,
  `electronicHealthRecord` INT NOT NULL ,
  `referringClinician` INT NOT NULL ,
  `referredClinician` INT NOT NULL ,
  `author` INT NOT NULL ,
  `creationTime` DATETIME NOT NULL ,
  `cancelled` TINYINT NOT NULL DEFAULT 0 ,
  `cancelledBy` INT NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Referral_EHRToEHRID`
    FOREIGN KEY (`electronicHealthRecord` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Referral_referringClinicianToClinicianID`
    FOREIGN KEY (`referringClinician` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Referral_referredClinicianToClinicianID`
    FOREIGN KEY (`referredClinician` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Referral_authorToUserID`
    FOREIGN KEY (`author` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Referral_cancelledByToUserID`
    FOREIGN KEY (`cancelledBy` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Referral_EHRToEHRID` ON `electronic_health_record_system`.`Referral` (`electronicHealthRecord` ASC) ;

CREATE INDEX `Referral_referringClinicianToClinicianID` ON `electronic_health_record_system`.`Referral` (`referringClinician` ASC) ;

CREATE INDEX `Referral_referredClinicianToClinicianID` ON `electronic_health_record_system`.`Referral` (`referredClinician` ASC) ;

CREATE INDEX `Referral_authorToUserID` ON `electronic_health_record_system`.`Referral` (`author` ASC) ;

CREATE INDEX `Referral_cancelledByToUserID` ON `electronic_health_record_system`.`Referral` (`cancelledBy` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Clinician`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Clinician` (
  `ID` INT NOT NULL ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Clinician_Medical_Areas`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Clinician_Medical_Areas` (
  `Clinician` INT NOT NULL ,
  `MedicalArea` INT NOT NULL ,
  PRIMARY KEY (`Clinician`, `MedicalArea`) ,
  CONSTRAINT `Clinician_Medical_Areas_ClinicianToClinicianID`
    FOREIGN KEY (`Clinician` )
    REFERENCES `electronic_health_record_system`.`Clinician` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Clinician_Medical_Areas_MedialAreaToMedicalAreaID`
    FOREIGN KEY (`MedicalArea` )
    REFERENCES `electronic_health_record_system`.`Medical_Areas` (`ID` )
    ON DELETE NO ACTION
```

```
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Clinician_Medical_Areas_ClinicianToClinicianID` ON `electronic_health_record_system`.`Clinician_Medical_Areas` (`Clinician` ASC) ;

CREATE INDEX `Clinician_Medical_Areas_MedialAreaToMedicalAreaID` ON `electronic_health_record_system`.`Clinician_Medical_Areas`
(`MedicalArea` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Treating_Clinicians`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Treating_Clinicians` (
  `electronicHealthRecord` INT NOT NULL ,
  `clinician` INT NOT NULL ,
  PRIMARY KEY (`electronicHealthRecord`, `clinician`) ,
  CONSTRAINT `Treating_Clinicians_EHRToEHRID`
    FOREIGN KEY (`electronicHealthRecord` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Treating_Clinicians_clinicianToClinicianID`
    FOREIGN KEY (`clinician` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Treating_Clinicians_EHRToEHRID` ON `electronic_health_record_system`.`Treating_Clinicians` (`electronicHealthRecord` ASC) ;

CREATE INDEX `Treating_Clinicians_clinicianToClinicianID` ON `electronic_health_record_system`.`Treating_Clinicians` (`clinician` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Appointed_Agents`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Appointed_Agents` (
  `agent` INT NOT NULL ,
  `electronicHealthRecord` INT NOT NULL ,
  `role` INT NOT NULL ,
  PRIMARY KEY (`agent`, `electronicHealthRecord`) ,
  CONSTRAINT `appointed_agents_agentToUserId`
    FOREIGN KEY (`agent` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `appointed_agents_EHRToEHRID`
    FOREIGN KEY (`electronicHealthRecord` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `appointed_agents_roleToRoleID`
    FOREIGN KEY (`role` )
    REFERENCES `electronic_health_record_system`.`Role` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `appointed_agents_agentToUserId` ON `electronic_health_record_system`.`Appointed_Agents` (`agent` ASC) ;

CREATE INDEX `appointed_agents_EHRToEHRID` ON `electronic_health_record_system`.`Appointed_Agents` (`electronicHealthRecord` ASC) ;

CREATE INDEX `appointed_agents_roleToRoleID` ON `electronic_health_record_system`.`Appointed_Agents` (`role` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Primary_Doctor`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Primary_Doctor` (
  `electronicHealthRecord` INT NOT NULL ,
  `primaryDoctor` INT NOT NULL ,
  PRIMARY KEY (`electronicHealthRecord`, `primaryDoctor`) ,
  CONSTRAINT `Primary_Doctor_EHRToEHRID`
    FOREIGN KEY (`electronicHealthRecord` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
```

```
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Primary_Doctor_primaryDoctorToClinicianID`
    FOREIGN KEY (`primaryDoctor` )
    REFERENCES `electronic_health_record_system`.`Clinician` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Primary_Doctor_EHRToEHRID` ON `electronic_health_record_system`.`Primary_Doctor` (`electronicHealthRecord` ASC) ;

CREATE INDEX `Primary_Doctor_primaryDoctorToClinicianID` ON `electronic_health_record_system`.`Primary_Doctor` (`primaryDoctor` ASC) ;

CREATE UNIQUE INDEX `uniqueEHR` USING HASH ON `electronic_health_record_system`.`Primary_Doctor` (`electronicHealthRecord` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Concealed_Item`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Concealed_Item` (
  `item` INT NOT NULL ,
  `blockedUser` INT NOT NULL ,
  PRIMARY KEY (`item`, `blockedUser`) ,
  CONSTRAINT `Concealed_Item_itemToItemID`
    FOREIGN KEY (`item` )
    REFERENCES `electronic_health_record_system`.`Comment_Item` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Concealed_Item_blockedUserToUserID`
    FOREIGN KEY (`blockedUser` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Concealed_Item_itemToItemID` ON `electronic_health_record_system`.`Concealed_Item` (`item` ASC) ;

CREATE INDEX `Concealed_Item_blockedUserToUserID` ON `electronic_health_record_system`.`Concealed_Item` (`blockedUser` ASC) ;


-- -----------------------------------------------------
-- Table `electronic_health_record_system`.`Log`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `electronic_health_record_system`.`Log` (
  `ID` INT NOT NULL ,
  `electronicHealthRecord` INT NULL ,
  `editingUser` INT NOT NULL ,
  `dateOfAction` DATETIME NOT NULL ,
  `actionDescription` VARCHAR(255) NOT NULL ,
  PRIMARY KEY (`ID`) ,
  CONSTRAINT `Log_EHRToEHRID`
    FOREIGN KEY (`electronicHealthRecord` )
    REFERENCES `electronic_health_record_system`.`Electronic_Health_Record` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Log_editingUserToUserID`
    FOREIGN KEY (`editingUser` )
    REFERENCES `electronic_health_record_system`.`User` (`ID` )
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

CREATE INDEX `Log_EHRToEHRID` ON `electronic_health_record_system`.`Log` (`electronicHealthRecord` ASC) ;

CREATE INDEX `Log_editingUserToUserID` ON `electronic_health_record_system`.`Log` (`editingUser` ASC) ;

USE `electronic_health_record_system`;

-- -----------------------------------------------------
-- Data for table `electronic_health_record_system`.`Role`
-- -----------------------------------------------------
SET AUTOCOMMIT=0;
INSERT INTO `Role` (`ID`, `RoleName`, `RoleDisplayName`, `SearchItemsInEHR`, `BrowseItemsInEHR`, `SearchAgents`, `BrowseAgents`,
`SearchSponsoredPatients`, `BrowseSponsoredPatients`, `SearchPatients`, `BrowsePatients`, `AppointAgent`, `RevokeAgent`, `RevokeCreatedAgent`,
`EditAgentPrivileges`, `AddQuantitativeObservationItem`, `AddQualitativeObservationItem`, `AddCommentItemToItem`, `AddCommentItem`,
```

```
`AddTreatmentItem`, `AddDiagnosisItem`, `ConcealItem`, `CancelItemConcealment`, `SetDotctorAsPrimary`, `RemoveDoctorAsPrimary`,
`GrantConsentToTreatment`, `WithdrawConsentToTreatment`, `AddUserToSystem`, `AddDoctorAsPrimary`, `RevokeDoctorAsPrimary`,
`CreateReferral`, `CancelReferral`, `CancelCreatedReferral`, `CreateMedicalArea`, `CreateObservedQuantityType`) VALUES (0, 'Patient', 'Patient', 1, 1,
1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0);
INSERT INTO `Role` (`ID`, `RoleName`, `RoleDisplayName`, `SearchItemsInEHR`, `BrowseItemsInEHR`, `SearchAgents`, `BrowseAgents`,
`SearchSponsoredPatients`, `BrowseSponsoredPatients`, `SearchPatients`, `BrowsePatients`, `AppointAgent`, `RevokeAgent`, `RevokeCreatedAgent`,
`EditAgentPrivileges`, `AddQuantitativeObservationItem`, `AddQualitativeObservationItem`, `AddCommentItemToItem`, `AddCommentItem`,
`AddTreatmentItem`, `AddDiagnosisItem`, `ConcealItem`, `CancelItemConcealment`, `SetDotctorAsPrimary`, `RemoveDoctorAsPrimary`,
`GrantConsentToTreatment`, `WithdrawConsentToTreatment`, `AddUserToSystem`, `AddDoctorAsPrimary`, `RevokeDoctorAsPrimary`,
`CreateReferral`, `CancelReferral`, `CancelCreatedReferral`, `CreateMedicalArea`, `CreateObservedQuantityType`) VALUES (1, 'Agent w/ Full', 'Agent',
1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0);
INSERT INTO `Role` (`ID`, `RoleName`, `RoleDisplayName`, `SearchItemsInEHR`, `BrowseItemsInEHR`, `SearchAgents`, `BrowseAgents`,
`SearchSponsoredPatients`, `BrowseSponsoredPatients`, `SearchPatients`, `BrowsePatients`, `AppointAgent`, `RevokeAgent`, `RevokeCreatedAgent`,
`EditAgentPrivileges`, `AddQuantitativeObservationItem`, `AddQualitativeObservationItem`, `AddCommentItemToItem`, `AddCommentItem`,
`AddTreatmentItem`, `AddDiagnosisItem`, `ConcealItem`, `CancelItemConcealment`, `SetDotctorAsPrimary`, `RemoveDoctorAsPrimary`,
`GrantConsentToTreatment`, `WithdrawConsentToTreatment`, `AddUserToSystem`, `AddDoctorAsPrimary`, `RevokeDoctorAsPrimary`,
`CreateReferral`, `CancelReferral`, `CancelCreatedReferral`, `CreateMedicalArea`, `CreateObservedQuantityType`) VALUES (2, 'Agent w/ Read',
'Agent', 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
INSERT INTO `Role` (`ID`, `RoleName`, `RoleDisplayName`, `SearchItemsInEHR`, `BrowseItemsInEHR`, `SearchAgents`, `BrowseAgents`,
`SearchSponsoredPatients`, `BrowseSponsoredPatients`, `SearchPatients`, `BrowsePatients`, `AppointAgent`, `RevokeAgent`, `RevokeCreatedAgent`,
`EditAgentPrivileges`, `AddQuantitativeObservationItem`, `AddQualitativeObservationItem`, `AddCommentItemToItem`, `AddCommentItem`,
`AddTreatmentItem`, `AddDiagnosisItem`, `ConcealItem`, `CancelItemConcealment`, `SetDotctorAsPrimary`, `RemoveDoctorAsPrimary`,
`GrantConsentToTreatment`, `WithdrawConsentToTreatment`, `AddUserToSystem`, `AddDoctorAsPrimary`, `RevokeDoctorAsPrimary`,
`CreateReferral`, `CancelReferral`, `CancelCreatedReferral`, `CreateMedicalArea`, `CreateObservedQuantityType`) VALUES (3, 'Agent w/ Read &
Comment', 'Agent', 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
INSERT INTO `Role` (`ID`, `RoleName`, `RoleDisplayName`, `SearchItemsInEHR`, `BrowseItemsInEHR`, `SearchAgents`, `BrowseAgents`,
`SearchSponsoredPatients`, `BrowseSponsoredPatients`, `SearchPatients`, `BrowsePatients`, `AppointAgent`, `RevokeAgent`, `RevokeCreatedAgent`,
`EditAgentPrivileges`, `AddQuantitativeObservationItem`, `AddQualitativeObservationItem`, `AddCommentItemToItem`, `AddCommentItem`,
`AddTreatmentItem`, `AddDiagnosisItem`, `ConcealItem`, `CancelItemConcealment`, `SetDotctorAsPrimary`, `RemoveDoctorAsPrimary`,
`GrantConsentToTreatment`, `WithdrawConsentToTreatment`, `AddUserToSystem`, `AddDoctorAsPrimary`, `RevokeDoctorAsPrimary`,
`CreateReferral`, `CancelReferral`, `CancelCreatedReferral`, `CreateMedicalArea`, `CreateObservedQuantityType`) VALUES (4, 'Clinician', 'Clinician', 1,
1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0);
INSERT INTO `Role` (`ID`, `RoleName`, `RoleDisplayName`, `SearchItemsInEHR`, `BrowseItemsInEHR`, `SearchAgents`, `BrowseAgents`,
`SearchSponsoredPatients`, `BrowseSponsoredPatients`, `SearchPatients`, `BrowsePatients`, `AppointAgent`, `RevokeAgent`, `RevokeCreatedAgent`,
`EditAgentPrivileges`, `AddQuantitativeObservationItem`, `AddQualitativeObservationItem`, `AddCommentItemToItem`, `AddCommentItem`,
`AddTreatmentItem`, `AddDiagnosisItem`, `ConcealItem`, `CancelItemConcealment`, `SetDotctorAsPrimary`, `RemoveDoctorAsPrimary`,
`GrantConsentToTreatment`, `WithdrawConsentToTreatment`, `AddUserToSystem`, `AddDoctorAsPrimary`, `RevokeDoctorAsPrimary`,
`CreateReferral`, `CancelReferral`, `CancelCreatedReferral`, `CreateMedicalArea`, `CreateObservedQuantityType`) VALUES (5, 'Administrator',
'Administrator', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0);

COMMIT;

-- -----------------------------------------------------
-- Data for table `electronic_health_record_system`.`Medical_Areas`
-- -----------------------------------------------------
SET AUTOCOMMIT=0;
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (1, 'Anesthesiology Critical Care Medicine');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (2, 'Bone Marrow Transplantation and Research Immunology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (3, 'Cardiac Critical Care');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (4, 'Cardiac Electrophysiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (5, 'Cardiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (6, 'Cardiothoracic Surgery');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (7, 'Celiac Disease');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (8, 'Child Abuse');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (9, 'Clinical Immunology/Allergy');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (10, 'Comfort and Pain Management');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (11, 'Communication Disorders');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (12, 'Cornea Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (13, 'Craniofacial & Cleft Care');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (14, 'Cystic Fibrosis');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (15, 'Dentistry');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (16, 'Dermatology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (17, 'Diagnostic Radiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (18, 'Dysmorphology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (19, 'Emergency Medicine');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (20, 'Endocrinology, Diabetes & Metabolism');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (21, 'Extra Corporeal Membrane Oxygenation');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (22, 'Gastroenterology and Nutrition');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (23, 'Fetal Echocardiography');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (24, 'General Pediatrics');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (25, 'Hand Surgery');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (26, 'Health Outcomes');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (27, 'Heart Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (28, 'Heart/Lung Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (29, 'Hematology/Oncology');
```

```
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (30, 'Hematology/Oncology (Bone & Extremity Tumor Program)');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (31, 'Hematology/Oncology (Neuro-Oncology Program)');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (32, 'Hemophilia');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (33, 'Hyperlipidemia/Hypercholesterolemia');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (34, 'Imaging Services');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (35, 'Immunology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (36, 'Infectious Diseases');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (37, 'Inflammatory Bowel Disease');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (38, 'Interventional Cardiac Catheterization');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (39, 'Interventional Radiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (40, 'Kawasaki Disease');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (41, 'Kidney Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (42, 'Liver Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (43, 'Lung Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (44, 'Magnetic Resonance Imaging');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (45, 'Maternal and Fetal Health');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (46, 'Medical Genetics');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (47, 'Metabolic Genetics');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (48, 'Mitochondrial Genetics');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (49, 'Neonatal Intensive Care');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (50, 'Nephrology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (51, 'Neurofibromatosis');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (52, 'Neurology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (53, 'Neuroradiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (54, 'Neurosurgery');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (55, 'Nuclear Imaging');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (56, 'Ophthalmology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (57, 'Orthopaedic Surgery');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (58, 'Otolaryngology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (59, 'Pathology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (60, 'Pediatric Anesthesiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (61, 'Pediatric Cardiovascular Intensive Care');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (62, 'Pediatric Intensive Care');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (63, 'Pediatric Radiology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (64, 'Pediatric Surgery');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (65, 'Plastic Surgery');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (66, 'Prenatal Genetics');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (67, 'Pulmonary Disease');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (68, 'Pulmonary Hypertension');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (69, 'Pulmonology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (70, 'Rehabilitation');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (71, 'Renal Dialysis and Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (72, 'Rheumatology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (73, 'Sickle Cell Disease');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (74, 'Sleep Disorders');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (75, 'Small Bowel Transplant');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (76, 'Spina Bifida');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (77, 'Sports Medicine');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (78, 'Telemedicine');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (79, 'Thoracic/Vascular Surgery (non-cardiac)');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (80, 'Total Parenteral Nutrition Service');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (81, 'Transesophageal Echocardiography');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (82, 'Trauma');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (83, 'Urology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (84, 'Virology');
INSERT INTO `Medical_Areas` (`ID`, `Description`) VALUES (85, 'Adolescent Medicine');

COMMIT;

-- -----------------------------------------------------
-- Data for table `electronic_health_record_system`.`Item_Types`
-- -----------------------------------------------------
SET AUTOCOMMIT=0;
INSERT INTO `Item_Types` (`ID`, `Description`) VALUES (1, 'Qualitative Observation');
INSERT INTO `Item_Types` (`ID`, `Description`) VALUES (2, 'Diagnosis');
INSERT INTO `Item_Types` (`ID`, `Description`) VALUES (3, 'Treatment');
INSERT INTO `Item_Types` (`ID`, `Description`) VALUES (4, 'Comment');
INSERT INTO `Item_Types` (`ID`, `Description`) VALUES (5, 'Quantitative Observation');

COMMIT;

-- -----------------------------------------------------
-- Data for table `electronic_health_record_system`.`Observed_Quantitity_Type`
-- -----------------------------------------------------
SET AUTOCOMMIT=0;
```

```
INSERT INTO `Observed_Quantitity_Type` (`ID`, `Description`) VALUES (1, 'Temperature');
INSERT INTO `Observed_Quantitity_Type` (`ID`, `Description`) VALUES (2, 'EKG');
INSERT INTO `Observed_Quantitity_Type` (`ID`, `Description`) VALUES (3, 'Height');
INSERT INTO `Observed_Quantitity_Type` (`ID`, `Description`) VALUES (4, 'Weight');
INSERT INTO `Observed_Quantitity_Type` (`ID`, `Description`) VALUES (5, 'Blood Pressure');

COMMIT;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```