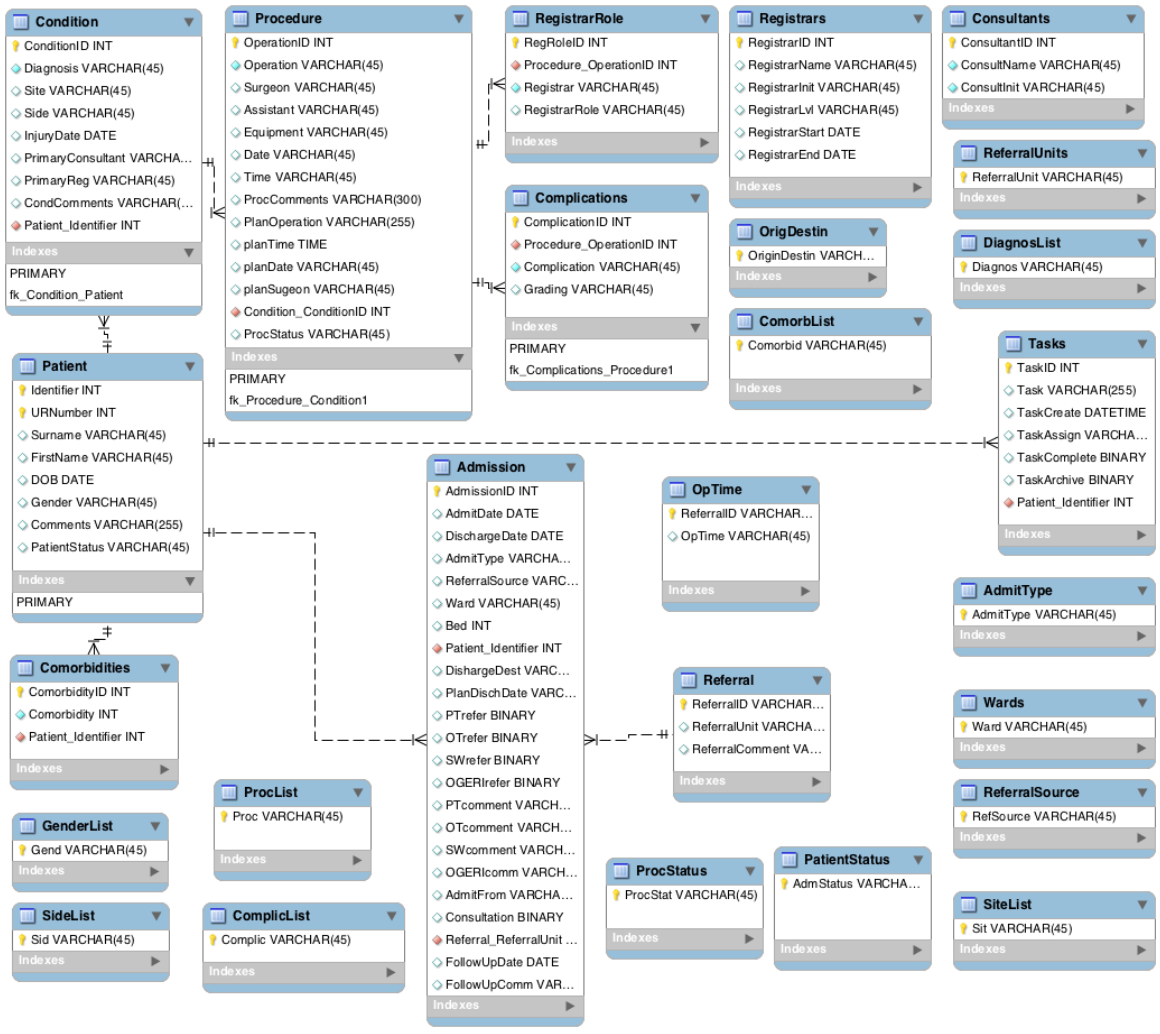
User Interfaces

The following user interfaces are required:

1. Edit Patient
2. Edit Condition
3. Edit Planned Procedure
3. Edit Completed Procedure
4. Edit Task

When a record is created in each , a new row should automatically be created in the relevant table (see MySQL Schema), with the relevant ***ID incremented to the next integer value, and the relevant foreign key selected to allow the record to be linked to the appropriate row in the “parent” table.

MySQL EER below for reference, MySQL code provided at end of document.



Edit Patient

UR No. [ ] Find

Finds if ID exists in Patient:URNumber. --> If Yes, populate with record
If No, create record and autoincrement Patient:Identifier

Surname [ ]

Patient:Surname (VarChar45)

First Name [ ]

Patient:FirstName (VarChar45)

D.O.B [ ]

Patient:DOB (Date)

Sex [ ]

Patient:Gender (picked from "Gender" Table: Male, Female)

Admission Status [ ]

Patient:PatientStatus (picked from "PatientStatus" Table: inpatient, Outpatient)

Comments [ ]

Patient:Comments

save changes + retrieve record

save changes + add new condition

1. Saves to table, and goes to "PatientViewRecord"
2. Saves to table, and
creates a "Condition" (autoincrement ConditionID, Patient_Identifier as Foreign Key)
goes to "EditCondition"

Edit Condition

PatientUR *Age* Yrs

Retreive from Patient

SURNAME, *Firstname*

Retreive from Patient

Diagnosis [ ]

Condition:Diagnosis (selects from list in DiagnosList)

Site [ ]

Condition:Site (selects from list in SiteList)

Side [ ]

Condition:Side (selects from list in SidList

Primary Reg [ ] Primary Consultant [ ]

Condition:PrimReg, Condition PrimConsult (select from table Registrars, Consultants)

Injury Date [ ]

Condition: Injury Date (DATE)

Comments [ ]

Condition:Comments (VarChar255)

save changes, go to patient record

save changes + add planned procedure

save changes + add completed procedure

1. Saves to table, goes to "PatientRecordView'

2. Saves to table,
creates a "Procedure" (autoincrements ProcedureID, Condition_ConditionID as foreign key)
goes to "EditPlanProcedure"

3. Saves to table,
creates a "Procedure" (autoincrements ProcedureID, Condition_ConditionID as foreign key)
goes to "EditProcedure"

Edit Planned Procedure

PatientUR *Age* Yrs

SURNAME, *Firstname*

Diagnosis *Site* *Side*

Plan Op

Plan Surgeon

Date Time

Equip

Comments

Status

save changes + return to record

convert to completed procedure

Retreive from Patient

Retreive from Patient

Retreive from Condition

Procedure:PlanOperation (VarChar255)

Procedure:PlanSurgeon

Procedure:PlanDate (DATE) Procedure:PlanTime (select from table ProcTime)

Procedure:Equipment (VarChar255)

Procedure:Comments (VarChar255)

Procedure:ProcStatus (default "pre-op" , select from table ProcStatus)

1. Saves to table, goes to "PatientRecordView"

2. Saves to table, goes to "OustandingCaseView"

3. Saves, and saves: PlanOperation--> Operation
PlanSurgeon-->Surgeon
PlanDate-->Date
PlanTime-->Time
Status--> "post-op"
Then goes to "EditCompletedProcedure"

Edit Completed Procedure

PatientUR *Age* Yrs

SURNAME, *Firstname*

Diagnosis *Site* *Side*

Operation

Surgeon

Assistant

Date Time

Equip

Comments

save changes + return to record

add another procedure for this condition

Retreive from Patient

Retreive from Patient

Retreive from Diagnosis

Procedure:Operation (selects from table ProcList)

Procedure:Surgeon

Procedure:Assistant

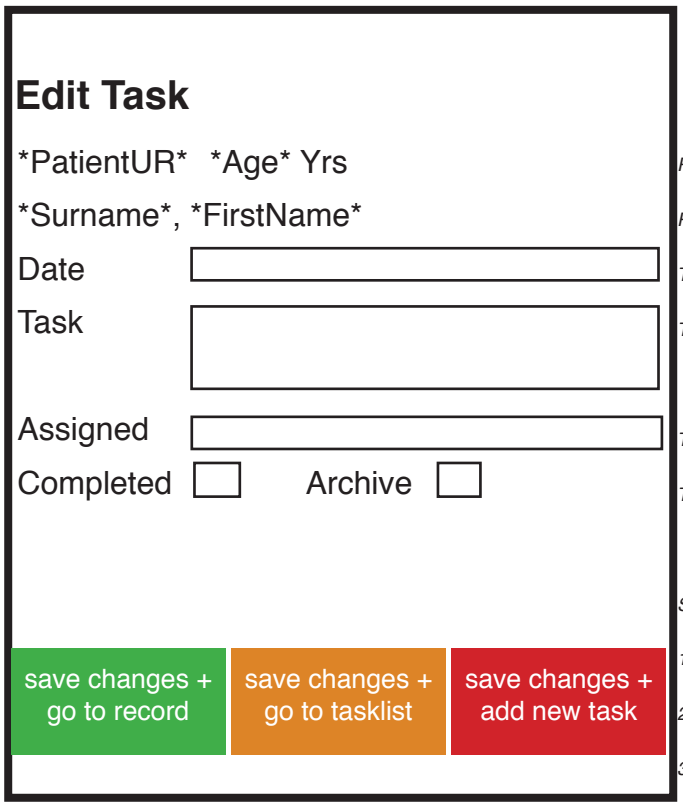
Procedure:Date (DATE) Procedure:Time (select from table ProcTime)

Procedure:Equipment (VarChar255)

Procedure:Comments (VarChar255)

Saves to table, then goes to "PatientRecordView"

Saves to table, then creates another "CompletedProcedure" for this Patient, Diagnosis



Retreive from table "Patient" , calculate "Age" based on "DOB"

Retreive from table "Patient"

Task:TaskCreate

Task:Task

Task:Assign (select from table from RegList)

Task:Complete (default is false) Task:Archive (default is false)

Saves to table, then goes to "PatientRecordView"

1. Saves to table, then goes to "PatientRecordView"

2. Saves to table, then goes to OustandingView

3. Saves to table, then adds new row in "Task" Table, with autoincremented Task:TaskID

```
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';

DROP SCHEMA IF EXISTS `mydb` ;
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1 COLLATE
latin1_swedish_ci ;
SHOW WARNINGS;
USE `mydb` ;

-- -----------------------------------------------------
-- Table `mydb`.`Patient`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Patient` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Patient` (
  `Identifier` INT NOT NULL AUTO_INCREMENT ,
  `URNumber` INT NOT NULL ,
  `Surname` VARCHAR(45) NULL ,
  `FirstName` VARCHAR(45) NULL ,
  `DOB` DATE NULL ,
  `Gender` VARCHAR(45) NULL ,
  `Comments` VARCHAR(255) NULL ,
  `PatientStatus` VARCHAR(45) NULL ,
  PRIMARY KEY (`Identifier`, `URNumber`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Condition`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Condition` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Condition` (
  `ConditionID` INT NOT NULL AUTO_INCREMENT ,
  `Diagnosis` VARCHAR(45) NOT NULL ,
  `Site` VARCHAR(45) NULL ,
  `Side` VARCHAR(45) NULL ,
  `InjuryDate` DATE NULL ,
  `PrimaryConsultant` VARCHAR(45) NULL ,
  `PrimaryReg` VARCHAR(45) NULL ,
  `CondComments` VARCHAR(300) NULL ,
  `Patient_Identifier` INT NOT NULL ,
  PRIMARY KEY (`ConditionID`) ,
  INDEX `fk_Condition_Patient` (`Patient_Identifier` ASC) ,
  CONSTRAINT `fk_Condition_Patient`
    FOREIGN KEY (`Patient_Identifier` )
    REFERENCES `mydb`.`Patient` (`Identifier` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Procedure`
```

```
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Procedure` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Procedure` (
  `OperationID` INT NOT NULL AUTO_INCREMENT ,
  `Operation` VARCHAR(45) NOT NULL ,
  `Surgeon` VARCHAR(45) NULL ,
  `Assistant` VARCHAR(45) NULL ,
  `Equipment` VARCHAR(45) NULL ,
  `Date` VARCHAR(45) NULL ,
  `Time` VARCHAR(45) NULL ,
  `ProcComments` VARCHAR(300) NULL ,
  `PlanOperation` VARCHAR(255) NULL ,
  `planTime` TIME NULL ,
  `planDate` VARCHAR(45) NULL ,
  `planSugeon` VARCHAR(45) NULL ,
  `Condition_ConditionID` INT NOT NULL ,
  `ProcStatus` VARCHAR(45) NULL ,
  PRIMARY KEY (`OperationID`) ,
  INDEX `fk_Procedure_Condition1` (`Condition_ConditionID` ASC) ,
  CONSTRAINT `fk_Procedure_Condition1`
    FOREIGN KEY (`Condition_ConditionID` )
    REFERENCES `mydb`.`Condition` (`ConditionID` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`RegistrarRole`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`RegistrarRole` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`RegistrarRole` (
  `RegRoleID` INT NOT NULL AUTO_INCREMENT ,
  `Procedure_OperationID` INT NOT NULL ,
  `Registrar` VARCHAR(45) NOT NULL ,
  `RegistrarRole` VARCHAR(45) NULL ,
  PRIMARY KEY (`RegRoleID`) ,
  INDEX `fk_RegistrarRole_Procedure1` (`Procedure_OperationID` ASC) ,
  CONSTRAINT `fk_RegistrarRole_Procedure1`
    FOREIGN KEY (`Procedure_OperationID` )
    REFERENCES `mydb`.`Procedure` (`OperationID` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Complications`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Complications` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Complications` (
```

```
  `ComplicationID` INT NOT NULL ,
  `Procedure_OperationID` INT NOT NULL ,
  `Complication` VARCHAR(45) NOT NULL ,
  `Grading` VARCHAR(45) NULL ,
  PRIMARY KEY (`ComplicationID`) ,
  INDEX `fk_Complications_Procedure1` (`Procedure_OperationID` ASC) ,
  CONSTRAINT `fk_Complications_Procedure1`
    FOREIGN KEY (`Procedure_OperationID` )
    REFERENCES `mydb`.`Procedure` (`OperationID` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Comorbidities`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Comorbidities` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Comorbidities` (
  `ComorbidityID` INT NOT NULL AUTO_INCREMENT ,
  `Comorbidity` INT NOT NULL ,
  `Patient_Identifier` INT NOT NULL ,
  PRIMARY KEY (`ComorbidityID`) ,
  INDEX `fk_Comorbidities_Patient1` (`Patient_Identifier` ASC) ,
  CONSTRAINT `fk_Comorbidities_Patient1`
    FOREIGN KEY (`Patient_Identifier` )
    REFERENCES `mydb`.`Patient` (`Identifier` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Referral`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Referral` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Referral` (
  `ReferralID` VARCHAR(45) NOT NULL ,
  `ReferralUnit` VARCHAR(45) NULL ,
  `ReferralComment` VARCHAR(355) NULL ,
  PRIMARY KEY (`ReferralID`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Admission`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Admission` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Admission` (
  `AdmissionID` INT NOT NULL AUTO_INCREMENT ,
```

```
  `AdmitDate` DATE NULL ,
  `DischargeDate` DATE NULL ,
  `AdmitType` VARCHAR(45) NULL ,
  `ReferralSource` VARCHAR(45) NULL ,
  `Ward` VARCHAR(45) NULL ,
  `Bed` INT NULL ,
  `Patient_Identifier` INT NOT NULL ,
  `DishargeDest` VARCHAR(45) NULL ,
  `PlanDischDate` VARCHAR(45) NULL ,
  `PTrefer` BINARY NULL ,
  `OTrefer` BINARY NULL ,
  `SWrefer` BINARY NULL ,
  `OGERIrefer` BINARY NULL ,
  `PTcomment` VARCHAR(45) NULL ,
  `OTcomment` VARCHAR(45) NULL ,
  `SWcomment` VARCHAR(45) NULL ,
  `OGERIcomm` VARCHAR(45) NULL ,
  `AdmitFrom` VARCHAR(45) NULL ,
  `Consultation` BINARY NULL ,
  `Referral_ReferralUnit` VARCHAR(45) NOT NULL ,
  `FollowUpDate` DATE NULL ,
  `FollowUpComm` VARCHAR(45) NULL ,
  PRIMARY KEY (`AdmissionID`) ,
  INDEX `fk_Admission_Patient1` (`Patient_Identifier` ASC) ,
  INDEX `fk_Admission_Referral1` (`Referral_ReferralUnit` ASC) ,
  CONSTRAINT `fk_Admission_Patient1`
    FOREIGN KEY (`Patient_Identifier` )
    REFERENCES `mydb`.`Patient` (`Identifier` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Admission_Referral1`
    FOREIGN KEY (`Referral_ReferralUnit` )
    REFERENCES `mydb`.`Referral` (`ReferralID` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Tasks`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Tasks` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Tasks` (
  `TaskID` INT NOT NULL AUTO_INCREMENT ,
  `Task` VARCHAR(255) NULL ,
  `TaskCreate` DATETIME NULL ,
  `TaskAssign` VARCHAR(45) NULL ,
  `TaskComplete` BINARY NULL ,
  `TaskArchive` BINARY NULL ,
  `Patient_Identifier` INT NOT NULL ,
  INDEX `fk_Tasks_Patient1` (`Patient_Identifier` ASC) ,
  PRIMARY KEY (`TaskID`) ,
  CONSTRAINT `fk_Tasks_Patient1`
    FOREIGN KEY (`Patient_Identifier` )
    REFERENCES `mydb`.`Patient` (`Identifier` )
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Registrars`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Registrars` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Registrars` (
  `RegistrarID` INT NOT NULL AUTO_INCREMENT ,
  `RegistrarName` VARCHAR(45) NULL ,
  `RegistrarInit` VARCHAR(45) NULL ,
  `RegistrarLvl` VARCHAR(45) NULL ,
  `RegistrarStart` DATE NULL ,
  `RegistrarEnd` DATE NULL ,
  PRIMARY KEY (`RegistrarID`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Consultants`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Consultants` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Consultants` (
  `ConsultantID` INT NOT NULL AUTO_INCREMENT ,
  `ConsultName` VARCHAR(45) NOT NULL ,
  `ConsultInit` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`ConsultantID`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`ComorbList`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`ComorbList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`ComorbList` (
  `Comorbid` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Comorbid`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`DiagnosList`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`DiagnosList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`DiagnosList` (
  `Diagnos` VARCHAR(45) NOT NULL ,
```

```
  PRIMARY KEY (`Diagnos`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`SiteList`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`SiteList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`SiteList` (
  `Sit` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Sit`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`ProcList`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`ProcList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`ProcList` (
  `Proc` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Proc`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`ComplicList`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`ComplicList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`ComplicList` (
  `Complic` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Complic`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`SideList`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`SideList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`SideList` (
  `Sid` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Sid`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`GenderList`
```

```
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`GenderList` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`GenderList` (
  `Gend` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Gend`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`PatientStatus`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`PatientStatus` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`PatientStatus` (
  `AdmStatus` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`AdmStatus`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`OrigDestin`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`OrigDestin` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`OrigDestin` (
  `OriginDestin` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`OriginDestin`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`ReferralUnits`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`ReferralUnits` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`ReferralUnits` (
  `ReferralUnit` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`ReferralUnit`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`AdmitType`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`AdmitType` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`AdmitType` (
  `AdmitType` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`AdmitType`) )
```

```
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`Wards`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`Wards` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`Wards` (
  `Ward` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Ward`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`ReferralSource`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`ReferralSource` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`ReferralSource` (
  `RefSource` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`RefSource`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`OpTime`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`OpTime` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`OpTime` (
  `ReferralID` VARCHAR(45) NOT NULL ,
  `OpTime` VARCHAR(45) NULL ,
  PRIMARY KEY (`ReferralID`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `mydb`.`ProcStatus`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`ProcStatus` ;

SHOW WARNINGS;
CREATE  TABLE IF NOT EXISTS `mydb`.`ProcStatus` (
  `ProcStat` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`ProcStat`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Placeholder table for view `mydb`.`view1`
```

```
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`view1` (`id` INT);
SHOW WARNINGS;

-- -----------------------------------------------------
-- View `mydb`.`view1`
-- -----------------------------------------------------
DROP VIEW IF EXISTS `mydb`.`view1` ;
SHOW WARNINGS;
DROP TABLE IF EXISTS `mydb`.`view1`;
SHOW WARNINGS;
USE `mydb`;
;
SHOW WARNINGS;


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