Trabajo Práctico Grupal

Administración de Reservas de Casos Especiales

1er. Cuatrimestre de 20112

75.15 Base de Datos

Facultad de Ingeniería, Universidad de Buenos Aires

Fecha de entrega: 2 de Mayo del 2012

Apellido y Nombre	Padrón	mail
Benez, Cristian		cbenez@gmail.com
Romera Ferrio, Fernando		fernandoromeraferrio@gmail.com
Scoppa, Alfredo	89149	alfredo.scoppa@hotmail.com
Szperling, Leonel		lszperling@gmail.com

Tabla de Contenidos

Enunciado del Trabajo Práctico Grupal.

Objetivo del Trabajo Práctico

1) Realizar un modelo MER en base al siguiente ERS y a los efectos de satisfacer los requerimientos de información solicitados.

2) Transformar el modelo E-R en un modelo relacional (modelo de tablas) utilizando los conocimientos de transformación de entidades a tablas.

Forma De Presentación Del Trabajo Practico

1) Presentar el diagrama de entidad - interrelación con indicaciones de restricciones de cardinalidad.

2) Indicar dependencias de identidad y de existencia en el modelo.

3) Especificar supuestos que justifiquen el modelo (Hipótesis).

4) Presentar el diagrama de entidad - interrelación con indicaciones de restricciones de cardinalidad.

Indicar dependencias de identidad y de existencia en el modelo.

Especificar supuestos que justifiquen el modelo (Hipótesis).

Presentar el diccionario de datos del diagrama con la siguiente información: Para cada tipo de entidad se debe especificar:

- Definición.
- Especificación de atributos.
- Especificación de identificador único.

Para cada tipo de interrelación se debe especificar:

- Definición.
- Especificación de atributos.
- Especificación de identificador único.

5) Presentar el modelo Relacional ("de tablas") indicando para cada

esquema de relación:

- Atributos
- Claves candidatas
- Clave primaria
- Claves foráneas
- Atributos que pueden tomar valores nulos
- Realice el diagrama del Modelo de Tablas
- Sentencias DDL

Nota: en los casos en que existan diferentes alternativas para efectuar la transformación de MER al modelo de tablas, elegir una única alternativa y enumerar las ventajas y desventajas de la alternativa elegida.

Diagrama Entidad - Interrelacíon

Hipótesis tomadas

Los supuestos considerados fueron:

Diccionario de Datos

Entidades

Entidad 1

Definición: Bla bla bla

Atributos:

- Atributo1
- Atributo2

Interrelaciones

Modelo Relacional

Los esquemas de relación son los que se listan a continuación. Las claves primarias son las subrayadas, las claves foráneas están indicadas en negrita y las candidatas que no son ni clave primaria ni clave foranea en cursiva.

Diagrama del Modelo de Tablas

Adjunto al final del documento.

Sentencias DDL

```
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 `grupo4` DEFAULT CHARACTER SET latin1
COLLATE latin1_swedish_ci ;
USE `grupo4` ;

-- -----------------------------------------------------
-- Table `grupo4`.`entidad_financiera`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`entidad_financiera` (
  `cod_entidad` INT NOT NULL ,
  `nombre` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`cod_entidad`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`plan_de_cobertura`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`plan_de_cobertura` (
  `cod_entidad` INT NOT NULL ,
  `numero_plan` INT NOT NULL ,
  `nombre` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`cod_entidad`, `numero_plan`) ,
  INDEX `fk_cod_entidad` (`cod_entidad` ASC) ,
  CONSTRAINT `fk_cod_entidad`
    FOREIGN KEY (`cod_entidad` )
    REFERENCES `grupo4`.`entidad_financiera` (`cod_entidad` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`medico`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`medico` (
```

```
  `matricula` INT NOT NULL ,
  `apellido` VARCHAR(100) NOT NULL ,
  `nombre` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`matricula`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`paciente`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`paciente` (
  `id_paciente` INT NOT NULL ,
  `nombre` VARCHAR(100) NOT NULL ,
  `apellido1` VARCHAR(100) NOT NULL ,
  `apellido2` VARCHAR(100) NOT NULL ,
  `tipo_documento` VARCHAR(100) NOT NULL ,
  `nro_documento` INT NOT NULL ,
  `condicion_iva` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`id_paciente`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`solicitud_tentativa`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`solicitud_tentativa` (
  `nro_solicitud` INT NOT NULL COMMENT '    ' ,
  `estado` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`nro_solicitud`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`tipo_procedimiento_medico`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`tipo_procedimiento_medico` (
  `codigo_procedimiento_medico` INT NOT NULL ,
  `nombre` VARCHAR(100) NOT NULL ,
  `duracion` INT NOT NULL ,
  PRIMARY KEY (`codigo_procedimiento_medico`) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `grupo4`.`indicacion_medica`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`indicacion_medica` (
  `nro_indicacion` INT NOT NULL ,
  `fecha_inicio` INT NOT NULL ,
  `hora_inicio` INT NOT NULL ,
  `tipo_anestesia` VARCHAR(100) NOT NULL ,
  `tiempo_anestesia` INT NOT NULL ,
  `matricula` INT NOT NULL ,
  `id_paciente` INT NOT NULL ,
  `nro_solicitud` INT NOT NULL ,
  `cod_cpt` INT NOT NULL ,
  PRIMARY KEY (`nro_indicacion`) ,
  INDEX `fk_matricula` (`matricula` ASC) ,
  INDEX `fk_id_paciente` (`id_paciente` ASC) ,
  INDEX `fk_nro_solicitud` (`nro_solicitud` ASC) ,
  INDEX `fk_cod_cpt` (`cod_cpt` ASC) ,
  CONSTRAINT `fk_matricula`
    FOREIGN KEY (`matricula` )
    REFERENCES `grupo4`.`medico` (`matricula` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_id_paciente`
    FOREIGN KEY (`id_paciente` )
    REFERENCES `grupo4`.`paciente` (`id_paciente` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_nro_solicitud`
    FOREIGN KEY (`nro_solicitud` )
    REFERENCES `grupo4`.`solicitud_tentativa` (`nro_solicitud` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_cod_cpt`
    FOREIGN KEY (`cod_cpt` )
    REFERENCES `grupo4`.`tipo_procedimiento_medico`
(`codigo_procedimiento_medico` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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

```
-- Table `grupo4`.`facturacion`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`facturacion` (
  `nro_facturacion` INT NOT NULL ,
  `monto` DOUBLE NOT NULL ,
  `nro_indicacion` INT NOT NULL ,
  `cod_entidad` INT NOT NULL ,
  `nro_plan` INT NOT NULL ,
  PRIMARY KEY (`nro_facturacion`) ,
  INDEX `fk_nro_indicacion` (`nro_indicacion` ASC) ,
--  INDEX `fk_cod_entidad_nro_plan` (`cod_entidad` ASC, `nro_plan` ASC) ,
  CONSTRAINT `fk_nro_indicacion`
    FOREIGN KEY (`nro_indicacion` )
    REFERENCES `grupo4`.`indicacion_medica` (`nro_indicacion` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
/*,
  CONSTRAINT `fk_cod_entidad_nro_plan`
    FOREIGN KEY (`cod_entidad` , `nro_plan` )
    REFERENCES `grupo4`.`plan_de_cobertura` (`cod_entidad` ,
`numero_plan` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION*/)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`historia_clinica`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`historia_clinica` (
  `nro_historia` INT NOT NULL ,
  `datos_paciente` VARCHAR(100) NOT NULL ,
  `id_paciente` INT NOT NULL ,
  PRIMARY KEY (`nro_historia`) ,
  INDEX `fk_id_paciente_historia_clinica` (`id_paciente` ASC) ,
  CONSTRAINT `fk_id_paciente_historia_clinica`
    FOREIGN KEY (`id_paciente` )
    REFERENCES `grupo4`.`paciente` (`id_paciente` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

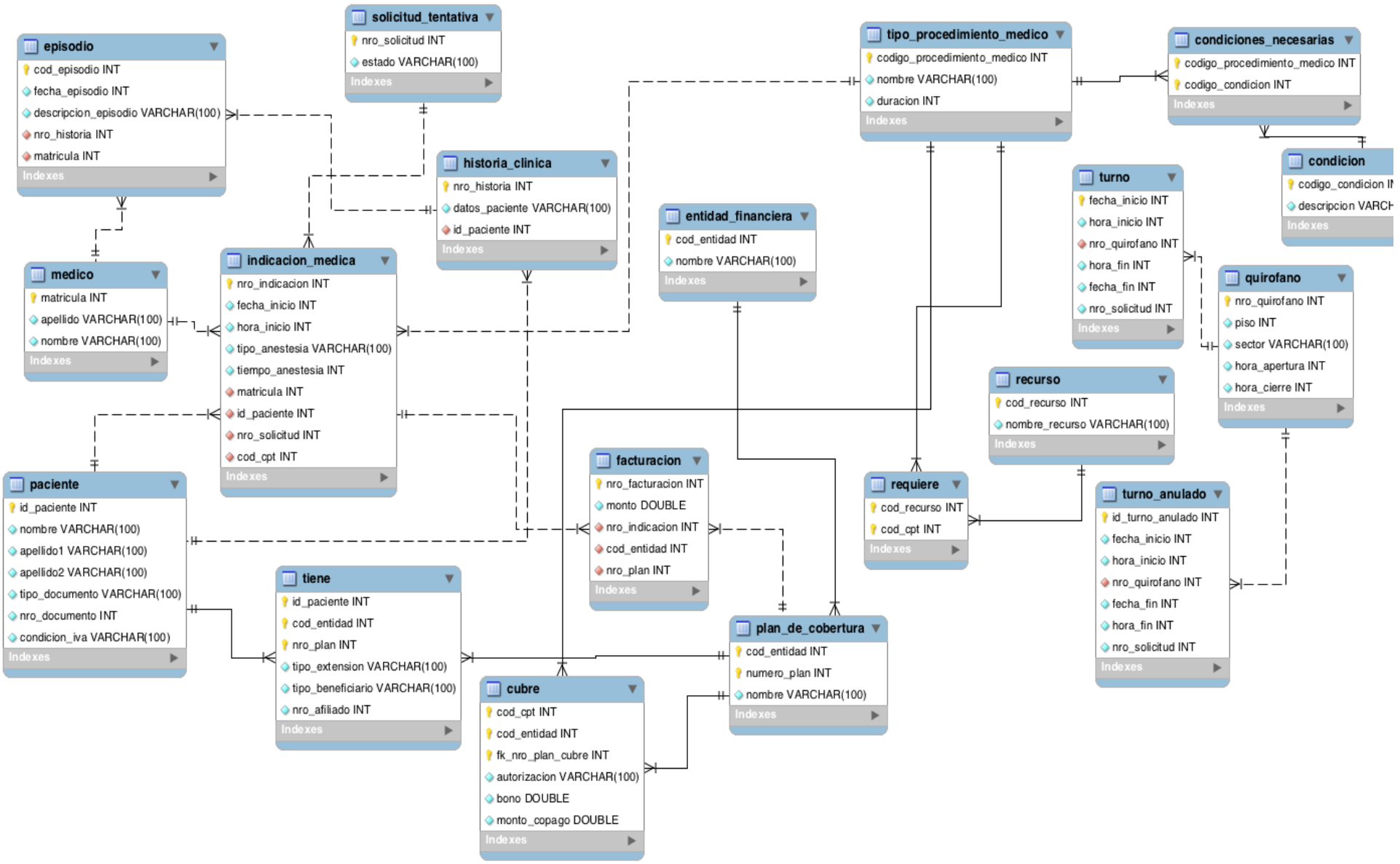
```
-- -----------------------------------------------------
-- Table `grupo4`.`episodio`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`episodio` (
  `cod_episodio` INT NOT NULL ,
  `fecha_episodio` INT NOT NULL ,
  `descripcion_episodio` VARCHAR(100) NOT NULL ,
  `nro_historia` INT NOT NULL ,
  `matricula` INT NOT NULL ,
  PRIMARY KEY (`cod_episodio`) ,
  INDEX `fk_nro_historia_episodio` (`nro_historia` ASC) ,
  INDEX `fk_matricula_episodio` (`matricula` ASC) ,
  CONSTRAINT `fk_nro_historia_episodio`
    FOREIGN KEY (`nro_historia` )
    REFERENCES `grupo4`.`historia_clinica` (`nro_historia` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_matricula_episodio`
    FOREIGN KEY (`matricula` )
    REFERENCES `grupo4`.`medico` (`matricula` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`recurso`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`recurso` (
  `cod_recurso` INT NOT NULL ,
  `nombre_recurso` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`cod_recurso`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`tiene`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`tiene` (
  `id_paciente` INT NOT NULL ,
  `cod_entidad` INT NOT NULL ,
  `nro_plan` INT NOT NULL ,
  `tipo_extension` VARCHAR(100) NOT NULL ,
```

```
  `tipo_beneficiario` VARCHAR(100) NOT NULL ,
  `nro_afiliado` INT NOT NULL ,
  PRIMARY KEY (`id_paciente`, `cod_entidad`, `nro_plan`) ,
  INDEX `fk_id_paciente_tiene` (`id_paciente` ASC) ,
--  INDEX `fk_cod_entidad_nro_plan_tiene` (`nro_plan` ASC, `cod_entidad`
ASC) ,
  CONSTRAINT `fk_id_paciente_tiene`
    FOREIGN KEY (`id_paciente` )
    REFERENCES `grupo4`.`paciente` (`id_paciente` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
/*
,
  CONSTRAINT `fk_cod_entidad_nro_plan_tiene`
    FOREIGN KEY (`nro_plan` , `cod_entidad` )
    REFERENCES `grupo4`.`plan_de_cobertura` (`numero_plan` ,
`cod_entidad` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
*/)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`cubre`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`cubre` (
  `cod_cpt` INT NOT NULL ,
  `cod_entidad` INT NOT NULL ,
  `nro_plan` INT NOT NULL ,
  `autorizacion` VARCHAR(100) NOT NULL ,
  `bono` DOUBLE NOT NULL ,
  `monto_copago` DOUBLE NOT NULL ,
  PRIMARY KEY (`cod_cpt`, `cod_entidad`, `nro_plan`) ,
  INDEX `fk_cod_cpt_cubre` (`cod_cpt` ASC) ,
--  INDEX `fk_cod_entidad_nro_plan_cubre` (`nro_plan` ASC, `cod_entidad`
ASC) ,

  CONSTRAINT `fk_cod_cpt_cubre`
    FOREIGN KEY (`cod_cpt` )
    REFERENCES `grupo4`.`tipo_procedimiento_medico`
(`codigo_procedimiento_medico` )
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION
/*
,
  CONSTRAINT `fk_cod_entidad_nro_plan_cubre`
    FOREIGN KEY (`nro_plan` , `cod_entidad` )
    REFERENCES `grupo4`.`plan_de_cobertura` (`numero_plan` ,
`cod_entidad` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
*/
)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`requiere`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`requiere` (
  `cod_recurso` INT NOT NULL ,
  `cod_cpt` INT NOT NULL ,
  PRIMARY KEY (`cod_recurso`, `cod_cpt`) ,
  INDEX `fk_cod_recurso` (`cod_recurso` ASC) ,
  INDEX `fk_cod_cpt_requiere` (`cod_cpt` ASC) ,
  CONSTRAINT `fk_cod_recurso`
    FOREIGN KEY (`cod_recurso` )
    REFERENCES `grupo4`.`recurso` (`cod_recurso` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_cod_cpt_requiere`
    FOREIGN KEY (`cod_cpt` )
    REFERENCES `grupo4`.`tipo_procedimiento_medico`
(`codigo_procedimiento_medico` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`quirofano`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`quirofano` (
  `nro_quirofano` INT NOT NULL ,
  `piso` INT NOT NULL ,
```

```
  `sector` VARCHAR(100) NOT NULL ,
  `hora_apertura` INT NOT NULL ,
  `hora_cierre` INT NOT NULL ,
  PRIMARY KEY (`nro_quirofano`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`turno_anulado`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`turno_anulado` (
  `id_turno_anulado` INT NOT NULL ,
  `fecha_inicio` INT NOT NULL ,
  `hora_inicio` INT NOT NULL ,
  `nro_quirofano` INT NOT NULL ,
  `fecha_fin` INT NOT NULL ,
  `hora_fin` INT NOT NULL ,
  `nro_solicitud` INT NOT NULL ,
  PRIMARY KEY (`id_turno_anulado`) ,
  INDEX `fk_nro_quirofano` (`nro_quirofano` ASC) ,
  CONSTRAINT `fk_nro_quirofano`
    FOREIGN KEY (`nro_quirofano` )
    REFERENCES `grupo4`.`quirofano` (`nro_quirofano` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`turno`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`turno` (
  `fecha_inicio` INT NOT NULL ,
  `hora_inicio` INT NOT NULL ,
  `nro_quirofano` INT NOT NULL ,
  `hora_fin` INT NOT NULL ,
  `fecha_fin` INT NOT NULL ,
  `nro_solicitud` INT NOT NULL ,
  PRIMARY KEY (`fecha_inicio`) ,
  INDEX `fk_nro_quirofano_turno` (`nro_quirofano` ASC) ,
  CONSTRAINT `fk_nro_quirofano_turno`
    FOREIGN KEY (`nro_quirofano` )
    REFERENCES `grupo4`.`quirofano` (`nro_quirofano` )
```

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


-- -----------------------------------------------------
-- Table `grupo4`.`condicion`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`condicion` (
  `codigo_condicion` INT NOT NULL ,
  `descripcion` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`codigo_condicion`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `grupo4`.`condiciones_necesarias`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `grupo4`.`condiciones_necesarias` (
  `codigo_procedimiento_medico` INT NOT NULL ,
  `codigo_condicion` INT NOT NULL ,
  PRIMARY KEY (`codigo_procedimiento_medico`, `codigo_condicion`) ,
  INDEX `fk_codigo_procedimiento_medico` (`codigo_procedimiento_medico`
ASC) ,
  INDEX `fk_codigo_condicion` (`codigo_condicion` ASC) ,
  CONSTRAINT `fk_codigo_procedimiento_medico`
    FOREIGN KEY (`codigo_procedimiento_medico` )
    REFERENCES `grupo4`.`tipo_procedimiento_medico`
(`codigo_procedimiento_medico` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_codigo_condicion`
    FOREIGN KEY (`codigo_condicion` )
    REFERENCES `grupo4`.`condicion` (`codigo_condicion` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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


episodio
cod_episodio INT
fecha_episodio INT
descripcion_episodio VARCHAR(100)
nro_historia INT
matricula INT
Indexes
solicitud_tentativa
nro_solicitud INT
estado VARCHAR(100)
Indexes
historia_clinica
nro_historia INT
datos_paciente VARCHAR(100)
id_paciente INT
Indexes
tipo_procedimiento_medico
codigo_procedimiento_medico INT
nombre VARCHAR(100)
duracion INT
Indexes
condiciones_necesarias
codigo_procedimiento_medico INT
codigo_condicion INT
Indexes
condicion
codigo_condicion IN
descripcion VARCH
Indexes
turno
fecha_inicio INT
hora_inicio INT
nro_quirofano INT
hora_fin INT
fecha_fin INT
nro_solicitud INT
Indexes
quirofano
nro_quirofano INT
piso INT
sector VARCHAR(100)
hora_apertura INT
hora_cierre INT
Indexes
entidad_financiera
cod_entidad INT
nombre VARCHAR(100)
Indexes
medico
matricula INT
apellido VARCHAR(100)
nombre VARCHAR(100)
Indexes
indicacion_medica
nro_indicacion INT
fecha_inicio INT
hora_inicio INT
tipo_anestesia VARCHAR(100)
tiempo_anestesia INT
matricula INT
id_paciente INT
nro_solicitud INT
cod_cpt INT
Indexes
recurso
cod_recurso INT
nombre_recurso VARCHAR(100)
Indexes
paciente
id_paciente INT
nombre VARCHAR(100)
apellido1 VARCHAR(100)
apellido2 VARCHAR(100)
tipo_documento VARCHAR(100)
nro_documento INT
condicion_iva VARCHAR(100)
Indexes
facturacion
nro_facturacion INT
monto DOUBLE
nro_indicacion INT
cod_entidad INT
nro_plan INT
Indexes
requiere
cod_recurso INT
cod_cpt INT
Indexes
turno_anulado
id_turno_anulado INT
fecha_inicio INT
hora_inicio INT
nro_quirofano INT
fecha_fin INT
hora_fin INT
nro_solicitud INT
Indexes
tiene
id_paciente INT
cod_entidad INT
nro_plan INT
tipo_extension VARCHAR(100)
tipo_beneficiario VARCHAR(100)
nro_afiliado INT
Indexes
plan_de_cobertura
cod_entidad INT
numero_plan INT
nombre VARCHAR(100)
Indexes
cubre
cod_cpt INT
cod_entidad INT
fk_nro_plan_cubre INT
autorizacion VARCHAR(100)
bono DOUBLE
monto_copago DOUBLE
Indexes