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 | 78783 | cbenez@gmail.com |
| Romera Ferrio, Fernando | 88406 | fernandoromeraferrio@gmail.com |
| Scoppa, Alfredo | 89149 | alfredo.scoppa@hotmail.com |
| Szperling, Leonel | 88845 | lszperling@gmail.com |

Tabla de Contenidos

Enunciado del Trabajo Práctico Grupal. 3

Objetivo del Trabajo Práctico  3

Forma De Presentación Del Trabajo Practico 3

Diagrama Entidad - Interrelacíon 5

Hipótesis tomadas 6

Diccionario de Datos 6

Entidades 6

Interrelaciones 8

Modelo Relacional 8

Diagrama del Modelo de Tablas 10

Sentencias DDL 11

# 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:

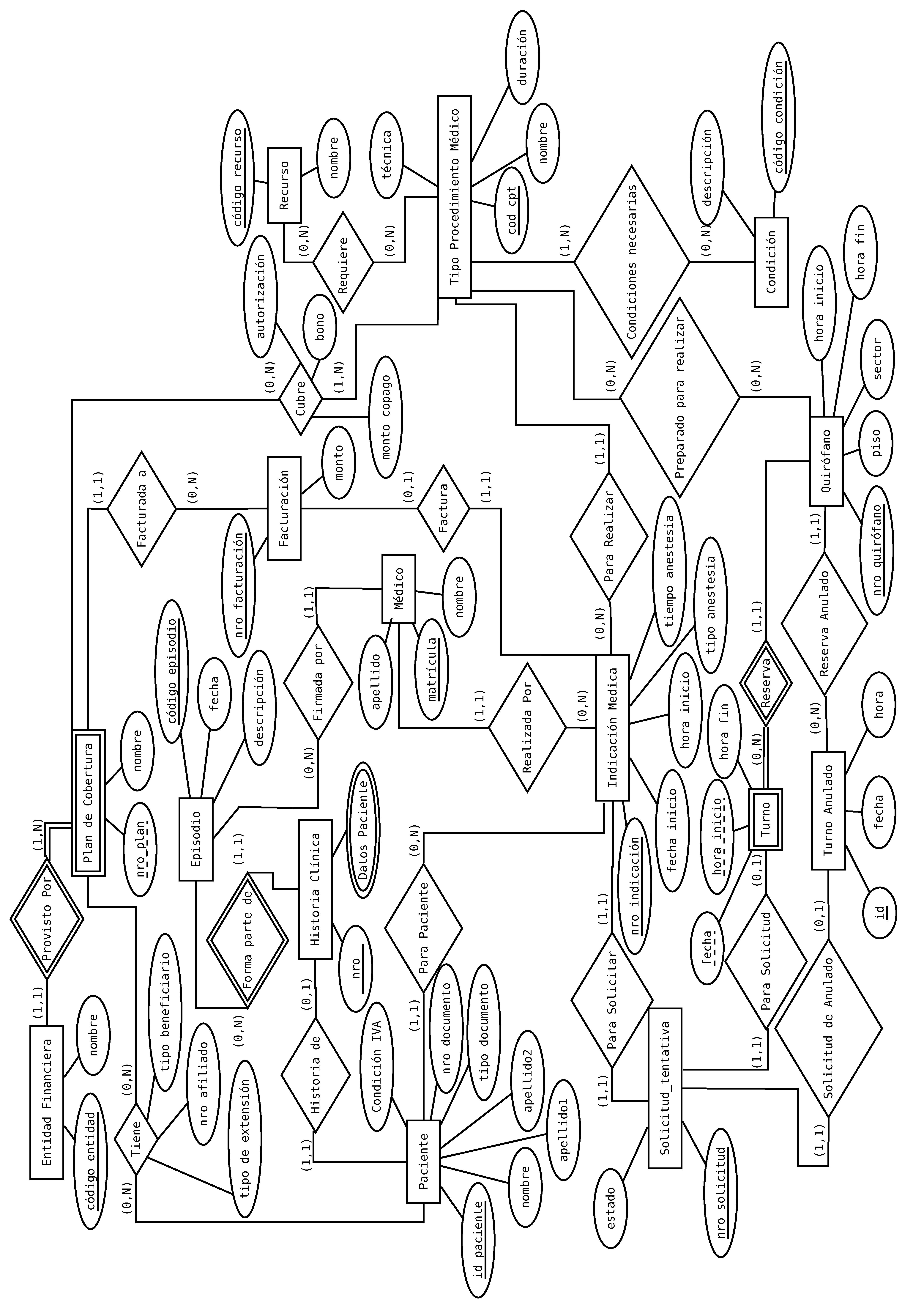
* Definición.
* Especificación de atributos.
* Especificación de identificador único.

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:

1. En caso de que un Paciente no tenga un Plan de Cobertura de una Entidad financiera correspondiente al procedimiento a realizar, se considerará una Entidad Financiera la cual será el mismo pacientey un Plan de Cobertura acorde.

2. Cada quirófano abre y cierra todos los días a la misma hora.

3. Tanto las Condiciones como los Recursos solicitados estan asociadas a los Tipos de Procedimiento Medico. Cada Tipo de Procedimiento Medico puede tener más de una Condición para que el paciente sea operable y solicitar más de un Recurso.

# Diccionario de Datos

## Entidades

CONDICION (codigo\_condicion,descripcion)

* Definicion: Representa una de las condiciones que debe cumplir un paciente para poder ser considerado para la intervencion
* Atributos: codigo\_condicion(identificador), descripcion(describe a la condicion)
* Identificador unico: codigo\_condicion

CONDICIONES\_NECESARIAS ()

* Definicion: Representa
* Atributos:
* Identificador unico:

ENTIDAD\_FINANCIERA ()

* Definicion: Representa
* Atributos:
* Identificador unico:

EPISODIO ()

* Definicion: Representa
* Atributos:
* Identificador unico:

FACTURACION ()

* Definicion: Representa
* Atributos:
* Identificador unico:

HISTORIA\_CLINICA ()

* Definicion: Representa
* Atributos:
* Identificador unico:

INDICACION\_MEDICA (codigo\_indicacion, fecha\_inicio, hora\_inicio, duracion\_estimada,tipo\_anestesia)

* Definicion: Representa una especificacion de la intervencion que debe realizarse sobre el paciente
* Atributos: codigo\_indicacion(identificador),fecha\_inicio(dia de inicio del procedimiento a efectuar),hora\_inicio(hora de inicio del procedimiento),duracion\_estimada(duracion promedio de la intervencion),tipo\_anestesia(describe el tipo de anestesia que se usara en la operacion)
* Identificador unico:codigo\_indicacion

MEDICO ()

* Definicion: Representa
* Atributos:
* Identificador unico:

PACIENTE ()

* Definicion: Representa
* Atributos:
* Identificador unico:

PLAN\_COBERTURA()

* Definicion: Representa
* Atributos:
* Identificador unico:

QUIROFANO()

* Definicion: Representa
* Atributos:
* Identificador unico:

RECURSO(codigo\_recurso,nombre)

* Definicion: Representa a los recursos usados durante la intervencion
* Atributos:codigo\_nombre (identificador), nombre (describe al recurso)
* Identificador unico:codigo\_recurso

SOLICITUD\_TENTATIVA()

* Definicion: Representa
* Atributos:
* Identificador unico:

TIPO\_PROCEDIMIENTO\_MEDICO(cod\_cpt,nombre,duracion,tecnica)

* Definicion: Representa al procedimiento quirurgico que se va a realizar sobre un paciente
* Atributos:
* Identificador unico:codigo\_cpt

TURNO()

* Definicion: Representa
* Atributos:
* Identificador unico:

TURNO\_ANULADO()

* Definicion: Representa
* Atributos:
* Identificador unico:

## 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; |

