

PLAN DE TRABAJO DEL ESTUDIANTE

1. INFORMACIÓN GENERAL

Apellidos y Nombres:	Romero Villanueva Sarai Rut	ID:	1472173
Dirección Zonal/CFP:	Huánuco		
Carrera:	Ingeniería de Software con I. A	Semestre:	II
Curso/ Mód. Formativo	202310-PIAD-214-TEC-NRC_39601		
Tema del Trabajo:	Modelo Físico de una Base de Datos Relacional		

2. PLANIFICACIÓN DEL TRABAJO

N°	ACTIVIDADES/ ENTREGABLES	CRONOGRAMA/ FECHA DE ENTREGA									
		1	8	/	0	5	/	2	0	2	3
1	Modelo Conceptual	1	8	/	0	5	/	2	0	2	3
2	Modelo Lógico	1	9	/	0	5	/	2	0	2	3
3	Modelo Físico	2	0	/	0	5	/	2	0	2	3
4	Un script de la Base de Datos	2	0	/	0	5	/	2	0	2	3
5											

3. PREGUNTAS GUIA

Durante la investigación de estudio, debes obtener las respuestas a las siguientes interrogantes:

Nº	PREGUNTAS
1	¿Cuáles son las diferencias entre un modelo conceptual, lógico y físico?
2	¿Qué es una Base de Datos Relacional?
3	¿Cuáles son los niveles de Normalización existentes en el modelamiento de BD?
4	¿Cuáles son las características de cada una las claves que existen en una entidad de una BD?
5	¿Cuáles son las diferencias entre el DDL y DML en SQL?

HOJA DE RESPUESTAS A LAS PREGUNTAS GUÍA

1.	<p>¿Cuáles son las diferencias entre un modelo conceptual, lógico y físico?</p> <p>Las diferencias entre el modelo conceptual y el modelo lógico y físico. Es que el modelo conceptual comienza con el análisis para sugerir una idea que funciona mediante la descripción de los datos y las relaciones entre las entidades y atributos.</p> <p>El modelo lógico es la recopilación de información e ideas para tenerla de formas más claras y organizadas.</p> <p>En cambio, el modelo físico entra con muchos más detalles, mostrando cómo se construirán los datos, y los diagramas de las relaciones entre las entidades y atributos mediante la conversión de la información en tablas y columnas.</p>
2.	<p>¿Qué es una Base de Datos Relacional?</p> <p>Una base de datos relacional es un conjunto de tablas formadas por filas y columnas; así, cada registro tiene una ID única, denominada clave y las columnas de la tabla contienen los atributos de los datos.</p>
3.	<p>¿Cuáles son los niveles de Normalización existentes en el modelamiento de BD?</p> <ul style="list-style-type: none"> Primera Forma Normal (1FN) Esta forma normal elimina los valores repetidos dentro de una base de datos. Segunda Forma Normal (2FN) La segunda forma normal está basada en el concepto de dependencia completamente funcional. Tercera Forma Normal (3FN) La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva en los atributos que no son clave. Forma normal de Boyce-Codd (FNBC) La tabla se encuentra en FNBC si cada determinante, atributo que determina completamente a otro, es clave candidata. Cuarta Forma Normal (4FN) Una tabla se encuentra en 4FN si, y solo si, para cada una de sus dependencias. multivaluadas no funcionales Quinta Forma Normal (5FN) La tabla está en 4FN y No existen relaciones de dependencias de reunión.
4.	<p>¿Cuáles son las características de cada una las claves que existen en una entidad de una BD?</p> <p>En función del estilo de implementación, los conceptos de entidad, atributo y tipo de entidad reflejan las prestaciones técnicas de MDM de tipo virtual y físico.</p>

El término registro de oro se suele emplear para describir el objetivo de proporcionar una vista de 360 grados de los datos maestros. Mientras que el término es suficientemente clarificador en conversaciones de alto nivel, las siguientes definiciones para conceptos que son más profundos ilustran cómo funcionan esos conceptos en InfoSphere MDM:

Entidad

Objeto único en el mundo real que se está controlando.

Tipo de entidad

Persona, organización, tipo de objeto o concepto sobre los que se almacena información.

Atributo

Característica o rasgo de un tipo de entidad que describe la entidad.

Registro

Representación del almacenamiento de una fila de datos.

Registro de miembro

Representación de la entidad tal como se almacena en los sistemas de origen individuales

5. ¿Cuáles son las diferencias entre el DDL y DML en SQL?

DDL es un lenguaje para describir los datos y sus relaciones en una base de datos. Puede generar DDL en un script de objetos de base de datos para: Mantener una instantánea de la estructura de base de datos. En cambio, DML (Lenguaje de Modificación de Datos) es muy importante en el lenguaje SQL. Lo forman las instrucciones capaces de añadir, cambiar o eliminar los datos de las tablas

6.



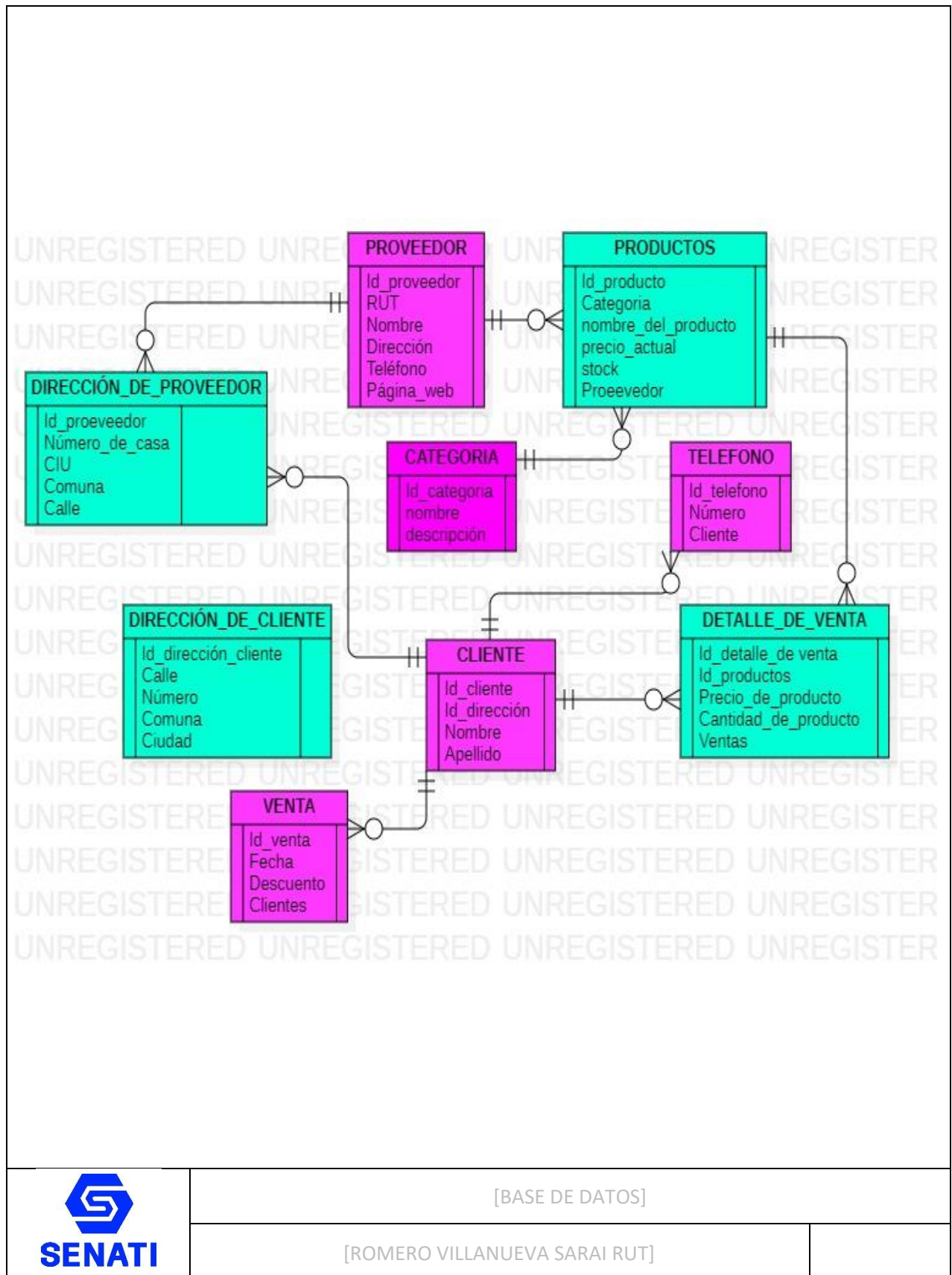
HOJA DE PLANIFICACIÓN

PROCESO DE EJECUCIÓN

[illegible]

INSTRUCCIONES: debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.

DIBUJO / ESQUEMA/ DIAGRAMA

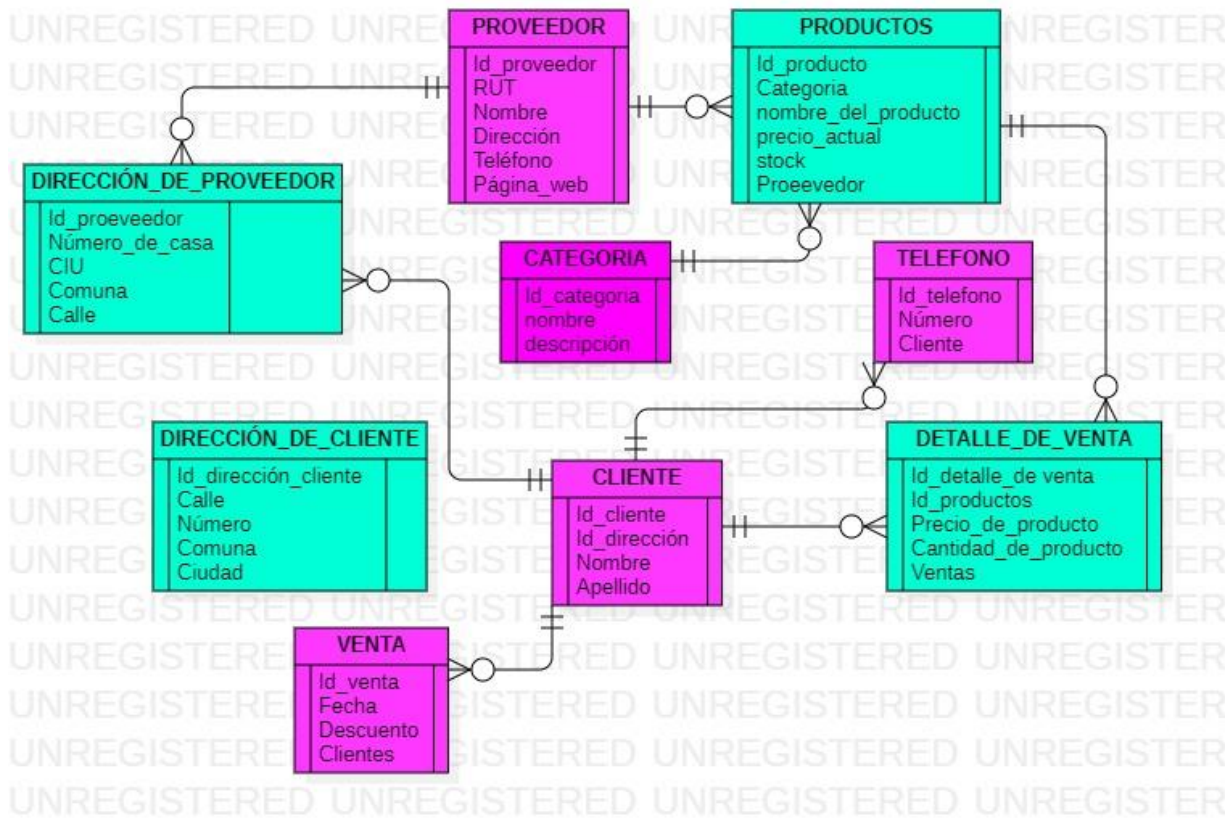


LISTA DE RECURSOS

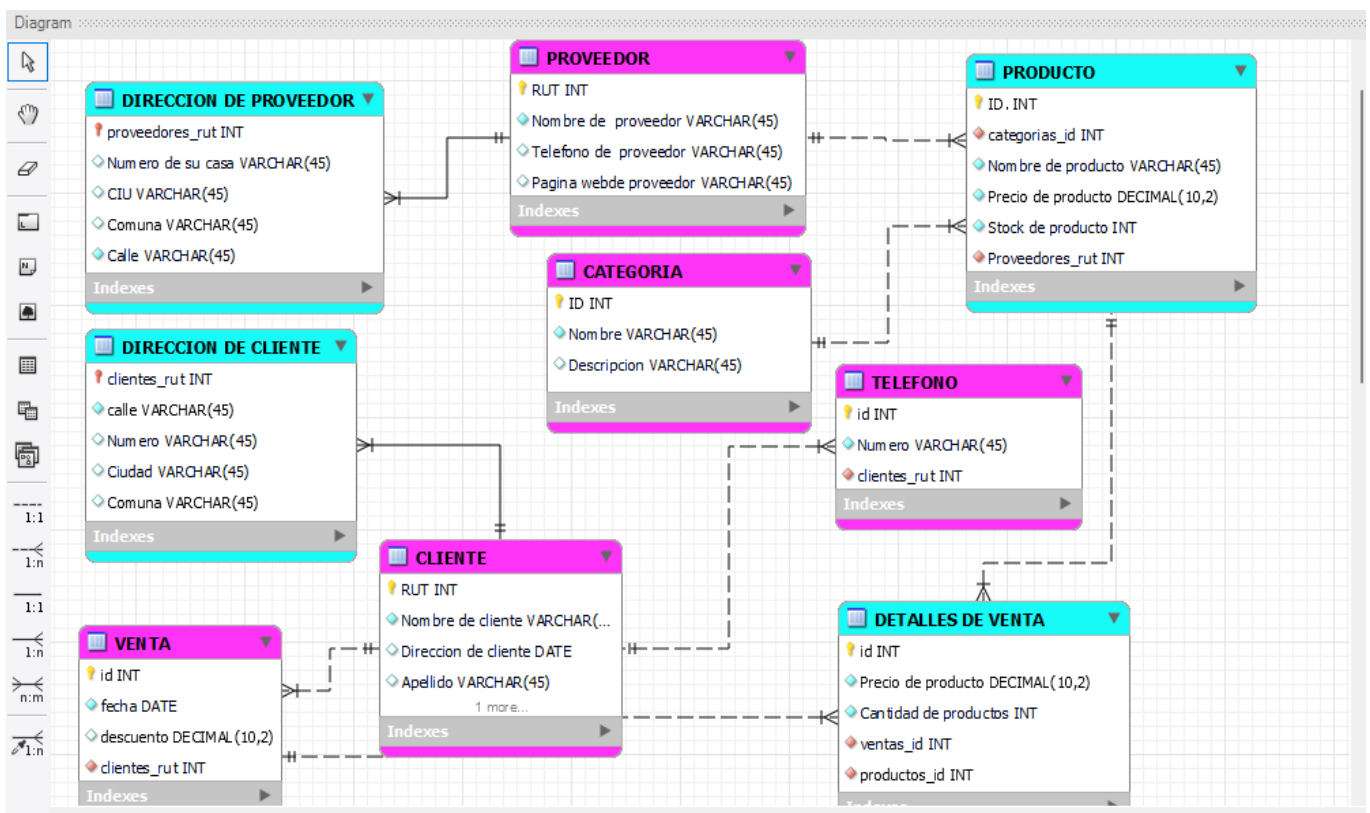
INSTRUCCIONES: completa la lista de recursos necesarios para la ejecución del trabajo.

1. MÁQUINAS Y EQUIPOS☒ Laptop☒ USB☒ Mouse☒ Celular**3. HERRAMIENTAS E INSTRUMENTOS**☒ MySQL Workbench☒ Plataforma ORACLE☒ Block de Notas☒ StarUML**5. MATERIALES E INSUMOS**☒ Internet☒ Bolígrafo☒ Cuaderno de apuntes

MÓDELO CONCEPTUAL:



MÓDELO LÓGICO:



**MÓDELO FISICO:**

```
Trabajo_final. x
-- BASE DE DATOS SOBRE VENTAS (PROYECTO FINAL)

1
2
3 • SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4 • SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5 • SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINEI
6
7 -----
8 • CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
9 • USE `mydb` ;
10
11 -----
12 -- Table `mydb`.`proveedor`
13 -----
14 • CREATE TABLE IF NOT EXISTS `mydb`.`proveedor` (
15   `rut` INT NOT NULL AUTO_INCREMENT,
16   `nombre` VARCHAR(45) NOT NULL,
17   `telefono` VARCHAR(45) NULL,
18   `pagina web` VARCHAR(45) NULL,
19   PRIMARY KEY (`rut`))
20   ENGINE = InnoDB;
21
Trabajo_final. x
22
23 -----
24 -- Table `mydb`.`clientes`
25 -----
26 • CREATE TABLE IF NOT EXISTS `mydb`.`cliente` (
27   `rut` INT NOT NULL AUTO_INCREMENT,
28   `nombre` VARCHAR(45) NOT NULL,
29   PRIMARY KEY (`rut`))
30   ENGINE = InnoDB;
31
32 -----
33 -- Table `mydb`.`categorias`
34 -----
35 -----
36 • CREATE TABLE IF NOT EXISTS `mydb`.`categoria` (
37   `id` INT NOT NULL AUTO_INCREMENT,
38   `nombre` VARCHAR(45) NOT NULL,
39   `descripcion` VARCHAR(45) NULL,
40   PRIMARY KEY (`id`))
41   ENGINE = InnoDB;
42
Trabajo_final. x
43
44 -----
45 -- Table `mydb`.`ventas`
46 -----
47 • CREATE TABLE IF NOT EXISTS `mydb`.`venta` (
48   `id` INT NOT NULL AUTO_INCREMENT,
49   `fecha` DATE NOT NULL,
50   `descuento` DECIMAL(10,2) NULL,
51   `clientes_rut` INT NOT NULL,
52   PRIMARY KEY (`id`),
53   INDEX `fk_ventas_clientes1_idx` (`clientes_rut` ASC) VISIBLE,
54   CONSTRAINT `fk_ventas_clientes1`
55     FOREIGN KEY (`clientes_rut`)
56     REFERENCES `mydb`.`clientes` (`rut`)
57     ON DELETE NO ACTION
58     ON UPDATE NO ACTION)
59   ENGINE = InnoDB;
60
```



```
Trabajo_final. x
Limit to 1000 rows

63  -- Table `mydb`.`productos`
64  -----
65  • CREATE TABLE IF NOT EXISTS `mydb`.`producto` (
66      `id` INT NOT NULL AUTO_INCREMENT,
67      `nombre` VARCHAR(45) NOT NULL,
68      `precio` DECIMAL(10,2) NOT NULL,
69      `stock` INT NOT NULL,
70      `categorias_id` INT NOT NULL,
71      `proveedores_rut` INT NOT NULL,
72      PRIMARY KEY (`id`),
73  ✖ INDEX `fk_productos_categorias_idx` (`categorias_id` ASC) VISIBLE,
74      INDEX `fk_productos_proveedores1_idx` (`proveedores_rut` ASC) VISIBLE,
75      CONSTRAINT `fk_productos_categorias`
76          FOREIGN KEY (`categorias_id`)
77              REFERENCES `mydb`.`categorias` (`id`)
78              ON DELETE NO ACTION
79              ON UPDATE NO ACTION,
80      CONSTRAINT `fk_productos_proveedores1`
81          FOREIGN KEY (`proveedores_rut`)
82              REFERENCES `mydb`.`proveedores` (`rut`)
83              ON DELETE NO ACTION
84              ON UPDATE NO ACTION)
85      ENGINE = InnoDB;
86  -----
87
88  -- Table `mydb`.`direccion_proveedor`
89  -----
90
91  • CREATE TABLE IF NOT EXISTS `mydb`.`direccion_proveedor` (
92      `proveedores_rut` INT NOT NULL,
93      `calle` VARCHAR(45) NOT NULL,
94      `numero` VARCHAR(45) NULL,
95      `ciudad` VARCHAR(45) NULL,
96      `comuna` VARCHAR(45) NULL,
97      PRIMARY KEY (`proveedores_rut`),
98      CONSTRAINT `fk_direccion_proveedor_proveedores1`
99          FOREIGN KEY (`proveedores_rut`)
100              REFERENCES `mydb`.`proveedores` (`rut`)
101              ON DELETE NO ACTION
102              ON UPDATE NO ACTION)
103      ENGINE = InnoDB;
104
105  -----
106
107  -- Table `mydb`.`direccion_cliente`
108  -----
109
110  • CREATE TABLE IF NOT EXISTS `mydb`.`direccion_cliente` (
111      `clientes_rut` INT NOT NULL,
112      `calle` VARCHAR(45) NOT NULL,
113      `numero` VARCHAR(45) NULL,
114      `ciudad` VARCHAR(45) NULL,
115      `comuna` VARCHAR(45) NULL,
116      PRIMARY KEY (`clientes_rut`),
117      CONSTRAINT `fk_direccion_cliente_clientes1`
118          FOREIGN KEY (`clientes_rut`)
119              REFERENCES `mydb`.`clientes` (`rut`)
120              ON DELETE NO ACTION
121              ON UPDATE NO ACTION)
122      ENGINE = InnoDB;
```



```
-- Table `mydb`.`telefonos`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`telefono` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `numero` VARCHAR(45) NOT NULL,
  `clientes_rut` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_telefonos_clientes1_idx` (`clientes_rut` ASC) VISIBLE,
  CONSTRAINT `fk_telefonos_clientes1`
    FOREIGN KEY (`clientes_rut`)
      REFERENCES `mydb`.`clientes` (`rut`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`detalle_venta`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`detalle_venta` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `precio` DECIMAL(10,2) NOT NULL,
  `cantidad` INT NOT NULL,
  `ventas_id` INT NOT NULL,
  `productos_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_detalle_venta_ventas1_idx` (`ventas_id` ASC) VISIBLE,
  INDEX `fk_detalle_venta_productos1_idx` (`productos_id` ASC) VISIBLE,
  CONSTRAINT `fk_detalle_venta_ventas1`
    FOREIGN KEY (`ventas_id`)
      REFERENCES `mydb`.`ventas` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_detalle_venta_productos1`
    FOREIGN KEY (`productos_id`)
      REFERENCES `mydb`.`productos` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
165
```

```
166 • SET SQL_MODE=@OLD_SQL_MODE;
167 • SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
168 • SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
169
```