

**TECNOLOGICO NACIONAL DE MEXICO**

**INSTITUTO TECNOLOGICO DE MORELIA**

**“José María Morelos y Pavón”**

**Taller de Base de Datos**

**Amaury Aurelio Coria Ramírez**

**Base de Datos Red Social FACEBOOK**

**Carlos Jahir Castro Cázares**

**Jaime Isaí Velázquez Aguilar**

**Ernesto Vieyra Melo**

**Víctor Hugo Ávila Pérez**

**Ingeniería en Sistemas Computacionales**

**Grupo: A**

**Semestre Ene-Jun 2019**

**04 de marzo del 2019**

***Tablas que poseen relaciones***

* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Lugares\_de\_trabajo (ID\_TRABAJO, TRABAJO)
* Relación: Perfil\_trabajo(ID\_PERFIL, ID\_TRABAJO, FECH\_INI, FECH\_FIN)
* Entidad Fuerte: Perfil
* Entidad Débil: Lugares\_de\_trabajo
* Cardinalidad: 1 a N, 1 Perfil N lugares
* Cardinalidad mínima: 1 perfil no requiere ningún lugar
* Participación total: No existe, 1 perfil no se relaciona con todos los lugares de trabajo
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Albumes (ID\_ALBUM, ID\_PERFIL, NOMBRE)
* Entidad Fuerte: Perfil
* Entidad Débil: Albumes
* Cardinalidad: 1 a N, 1 Perfil N albumes
* Cardinalidad mínima: 1 perfil no requiere ningún álbum
* Participación total: No existe, 1 perfil no se relaciona con todos los álbumes
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Titulos\_academicos (ID\_TITULO, TITULO)
* Relación: Perfil\_titulo(ID\_PERFIL, ID\_TITULO)
* Entidad Fuerte: Perfil
* Entidad Débil: Titulos\_academicos
* Cardinalidad: 1 a N, 1 Perfil N títulos
* Cardinalidad mínima: 1 perfil no requiere ningún titulo
* Participación total: No existe, 1 perfil no se relaciona con todos los títulos
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Certificación (ID\_CERTIFICACION, CERTIFICACION)
* Relación: Perfil\_certifi (ID\_PERFIL, ID\_CERTIFICACION)
* Entidad Fuerte: Perfil
* Entidad Débil: Certificación
* Cardinalidad: 1 a N, 1 Perfil N certificados
* Cardinalidad mínima: 1 perfil no requiere ningún certificado
* Participación total: No existe, 1 perfil no se relaciona con todos los certificados
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Películas (ID\_PELICULAS, PELICULA)
* Relación: Peliculas\_fav (ID\_PERFIL, ID\_PELICULA)
* Entidad Fuerte: Perfil
* Entidad Débil: Películas
* Cardinalidad: 1 a N, 1 Perfil N películas
* Cardinalidad mínima: 1 perfil no requiere ninguna película
* Participación total: No existe, 1 perfil no se relaciona con todas las películas
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Cantantes (ID\_CANTANTE, CANTANTE)
* Relación: Cantantes\_fav (ID\_PERFIL, ID\_CANTANTE)
* Entidad Fuerte: Perfil
* Entidad Débil: Cantantes
* Cardinalidad: 1 a N, 1 Perfil N cantantes
* Cardinalidad mínima: 1 perfil no requiere ningún cantante
* Participación total: No existe, 1 perfil no se relaciona con todos los cantantes
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Libros (ID\_LIBROS, LIBRO)
* Relación: Libros\_fav (ID\_PERFIL, ID\_LIBROS)
* Entidad Fuerte: Perfil
* Entidad Débil: Libros
* Cardinalidad: 1 a N, 1 Perfil N libros
* Cardinalidad mínima: 1 perfil no requiere ningún libro
* Participación total: No existe, 1 perfil no se relaciona con todos los libros
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Amigos (ID\_PERFIL, ID\_AMIGO)
* Entidad Fuerte: Perfil
* Entidad Débil: Perfil
* Cardinalidad: N a N, Perfil N amigos
* Cardinalidad mínima: 1 perfil no requiere ningún amigo
* Participación total: No existe, 1 perfil no se relaciona con todos los perfiles (Amigo)
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Historial (ID\_HISTORIAL, ID\_PERFIL, TEXT, IMAGEN, LIGA)
* Entidad Fuerte: Perfil
* Entidad Débil: Historial
* Cardinalidad: 1 a 1, Perfil 1 Historial
* Cardinalidad mínima: 1 perfil requiere 1 historial
* Participación total: No existe, 1 perfil solo se relaciona con un historial
* Historial (ID\_HISTORIAL, ID\_PERFIL, TEXT, IMAGEN, LIGA)
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Entidad Fuerte: Historial
* Entidad Débil: Publicacion
* Cardinalidad: 1 a N, 1 Historial N Publicacion
* Cardinalidad mínima: 1 Historial no requiere ninguna publicacion
* Participación total: No existe, un historial no se relaciona con todas las publicaciones y una publicación no se relaciona con todos los historiales
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Ubicación (ID\_UBICACION, UBICACION)
* Entidad Fuerte: Ubicacion
* Entidad Débil: Publicacion
* Cardinalidad: N a 1, N Publicaciones tienen 1 Publicacion
* Cardinalidad mínima: 1 Publicacion requiere 1 Ubicación
* Participación total: No existe, una publicación solo requiere una ubicación
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Estado (ID\_ESTADO, ESTADO)
* Entidad Fuerte: Estado
* Entidad Débil: Publicacion
* Cardinalidad: N a 1, N Publicaciones tienen 1 Estado
* Cardinalidad mínima: 1 Publicacion requiere un Estado
* Participación total: No existe, una publicación solo requiere un Estado
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Relación: Me\_Gusta (ID\_PERFIL, ID\_PUBLICACION)
* Entidad Fuerte: Publicacion
* Entidad Débil: Perfil
* Cardinalidad: N a N, N publicaciones pueden tener N perfiles
* Cardinalidad mínima: 1 Publicacion no requiere ningún perfil (Me gusta)
* Participación total: Si ya que una publicación puede tener todos los perfiles con me gusta
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Relación: Etiquetado(ID\_PERFIL, ID\_PUBLICACION)
* Entidad Fuerte: Publicacion
* Entidad Débil: Perfil
* Cardinalidad: N a N, N publicaciones pueden tener N perfiles
* Cardinalidad mínima: 1 Publicacion no requiere ningún perfil (Etiquetado)
* Participación total: Si ya que una publicación puede tener todos los perfiles etiquetados
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Amigos (ID\_PERFIL, ID\_AMIGO)
* Entidad Fuerte: Perfil
* Entidad Débil: Perfil
* Cardinalidad: N a N, Perfil N amigos
* Cardinalidad mínima: 1 perfil no requiere ningún amigo
* Participación total: No existe, 1 perfil no se relaciona con todos los perfiles (Amigo)
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Historial (ID\_HISTORIAL, ID\_PERFIL, TEXT, IMAGEN, LIGA)
* Entidad Fuerte: Perfil
* Entidad Débil: Historial
* Cardinalidad: 1 a 1, Perfil 1 Historial
* Cardinalidad mínima: 1 perfil requiere 1 historial
* Participación total: No existe, 1 perfil solo se relaciona con un historial
* Historial (ID\_HISTORIAL, ID\_PERFIL, TEXT, IMAGEN, LIGA)
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Entidad Fuerte: Historial
* Entidad Débil: Publicacion
* Cardinalidad: 1 a N, 1 Historial N Publicacion
* Cardinalidad mínima: 1 Historial no requiere ninguna publicacion
* Participación total: No existe, un historial no se relaciona con todas las publicaciones y una publicación no se relaciona con todos los historiales
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Ubicación (ID\_UBICACION, UBICACION)
* Entidad Fuerte: Ubicacion
* Entidad Débil: Publicacion
* Cardinalidad: N a 1, N Publicaciones tienen 1 Publicacion
* Cardinalidad mínima: 1 Publicacion requiere 1 Ubicación
* Participación total: No existe, una publicación solo requiere una ubicación
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Estado (ID\_ESTADO, ESTADO)
* Entidad Fuerte: Estado
* Entidad Débil: Publicacion
* Cardinalidad: N a 1, N Publicaciones tienen 1 Estado
* Cardinalidad mínima: 1 Publicacion requiere un Estado
* Participación total: No existe, una publicación solo requiere un Estado
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Relación: Me\_Gusta (ID\_PERFIL, ID\_PUBLICACION)
* Entidad Fuerte: Publicacion
* Entidad Débil: Perfil
* Cardinalidad: N a N, N publicaciones pueden tener N perfiles
* Cardinalidad mínima: 1 Publicacion no requiere ningún perfil (Me gusta)
* Participación total: Si ya que una publicación puede tener todos los perfiles con me gusta
* Publicacion (ID\_PUBLICACION, ID\_HISTORIAL, TEXTO, DIA-HORA, FOTO, ID\_UBICACION, ID\_ESTADO)
* Perfil (ID\_PERFIL, NOMBRE, CORREO, FECH\_NACI)
* Relación: Etiquetado(ID\_PERFIL, ID\_PUBLICACION)
* Entidad Fuerte: Publicacion
* Entidad Débil: Perfil
* Cardinalidad: N a N, N publicaciones pueden tener N perfiles
* Cardinalidad mínima: 1 Publicacion no requiere ningún perfil (Etiquetado)
* Participación total: Si ya que una publicación puede tener todos los perfiles etiquetados

***Código LDD***

-- MySQL Workbench Forward Engineering

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,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema BD\_Facebook

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema BD\_Facebook

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `BD\_Facebook` DEFAULT CHARACTER SET utf8mb4 ;

USE `BD\_Facebook` ;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Titulos\_Acacdemicos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Titulos\_Acacdemicos` (

`ID\_TITULO` INT NOT NULL AUTO\_INCREMENT,

`TITULO` VARCHAR(50) NOT NULL,

PRIMARY KEY (`ID\_TITULO`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Lugar\_Trabajo`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Lugar\_Trabajo` (

`ID\_LUGAR-TRABAJO` INT NOT NULL AUTO\_INCREMENT,

`TRABAJO` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ID\_LUGAR-TRABAJO`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Certificacion`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Certificacion` (

`ID\_CERTIFICACION` INT NOT NULL AUTO\_INCREMENT,

`CERTIFICACION` VARCHAR(50) NOT NULL,

PRIMARY KEY (`ID\_CERTIFICACION`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Peliculas`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Peliculas` (

`ID-PELICULA` INT NOT NULL AUTO\_INCREMENT,

`PELICULA` VARCHAR(100) NOT NULL,

PRIMARY KEY (`ID-PELICULA`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Cantante`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Cantante` (

`ID\_CANTANTE` INT NOT NULL AUTO\_INCREMENT,

`CANTANTE` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ID\_CANTANTE`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Libro`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Libro` (

`ID\_LIBRO` INT NOT NULL AUTO\_INCREMENT,

`LIBRO` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ID\_LIBRO`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Perfil`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Perfil` (

`ID\_PERFIL` INT NOT NULL AUTO\_INCREMENT,

`NOMBRE` VARCHAR(45) NOT NULL,

`CORREO` VARCHAR(45) NOT NULL,

`FECH\_NACI` DATE NOT NULL,

PRIMARY KEY (`ID\_PERFIL`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Estado`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Estado` (

`ID\_ESTADO` INT NOT NULL AUTO\_INCREMENT,

`ESTADO` VARCHAR(40) NOT NULL,

PRIMARY KEY (`ID\_ESTADO`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Ubicacion`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Ubicacion` (

`ID\_UBICACION` INT NOT NULL AUTO\_INCREMENT,

`UBICACION` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ID\_UBICACION`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Perfil-Titulos\_Acacdemicos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Perfil-Titulos\_Acacdemicos` (

`ID\_PERFIL` INT NOT NULL,

`ID\_TITULO` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Titulos\_Acacdemicos\_Titulos\_Acacdemicos1\_idx` (`ID\_TITULO` ASC),

INDEX `fk\_Perfil\_has\_Titulos\_Acacdemicos\_Perfil\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Titulos\_Acacdemicos\_Perfil`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Titulos\_Acacdemicos\_Titulos\_Acacdemicos1`

FOREIGN KEY (`ID\_TITULO`)

REFERENCES `BD\_Facebook`.`Titulos\_Acacdemicos` (`ID\_TITULO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Perfil-Lugar\_Trabajo`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Perfil-Lugar\_Trabajo` (

`ID\_PERFIL` INT NOT NULL,

`ID\_LUGAR-TRABAJO` INT NOT NULL,

`FECHA-FIN` DATE NOT NULL,

`FECHA-INI` DATE NOT NULL,

INDEX `fk\_Perfil\_has\_Lugar\_Trabajo\_Lugar\_Trabajo1\_idx` (`ID\_LUGAR-TRABAJO` ASC),

INDEX `fk\_Perfil\_has\_Lugar\_Trabajo\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Lugar\_Trabajo\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Lugar\_Trabajo\_Lugar\_Trabajo1`

FOREIGN KEY (`ID\_LUGAR-TRABAJO`)

REFERENCES `BD\_Facebook`.`Lugar\_Trabajo` (`ID\_LUGAR-TRABAJO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Perfil-Certificacion`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Perfil-Certificacion` (

`ID\_PERFIL` INT NOT NULL,

`ID\_CERTIFICACION` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Certificacion\_Certificacion1\_idx` (`ID\_CERTIFICACION` ASC),

INDEX `fk\_Perfil\_has\_Certificacion\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Certificacion\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Certificacion\_Certificacion1`

FOREIGN KEY (`ID\_CERTIFICACION`)

REFERENCES `BD\_Facebook`.`Certificacion` (`ID\_CERTIFICACION`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Fav-Peliculas`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Fav-Peliculas` (

`ID\_PERFIL` INT NOT NULL,

`ID-PELICULA` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Peliculas\_Peliculas1\_idx` (`ID-PELICULA` ASC),

INDEX `fk\_Perfil\_has\_Peliculas\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Peliculas\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Peliculas\_Peliculas1`

FOREIGN KEY (`ID-PELICULA`)

REFERENCES `BD\_Facebook`.`Peliculas` (`ID-PELICULA`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Fav-Cantante`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Fav-Cantante` (

`ID\_PERFIL` INT NOT NULL,

`ID\_CANTANTE` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Cantante\_Cantante1\_idx` (`ID\_CANTANTE` ASC),

INDEX `fk\_Perfil\_has\_Cantante\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Cantante\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Cantante\_Cantante1`

FOREIGN KEY (`ID\_CANTANTE`)

REFERENCES `BD\_Facebook`.`Cantante` (`ID\_CANTANTE`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Fav-Libro`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Fav-Libro` (

`ID\_PERFIL` INT NOT NULL,

`ID\_LIBRO` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Libro\_Libro1\_idx` (`ID\_LIBRO` ASC),

INDEX `fk\_Perfil\_has\_Libro\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Libro\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Libro\_Libro1`

FOREIGN KEY (`ID\_LIBRO`)

REFERENCES `BD\_Facebook`.`Libro` (`ID\_LIBRO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Album`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Album` (

`ID\_ALBUM` INT NOT NULL AUTO\_INCREMENT,

`NOMBRE` VARCHAR(45) NOT NULL,

`Perfil\_ID\_PERFIL` INT NOT NULL,

PRIMARY KEY (`ID\_ALBUM`),

INDEX `fk\_Album\_Perfil1\_idx` (`Perfil\_ID\_PERFIL` ASC),

CONSTRAINT `fk\_Album\_Perfil1`

FOREIGN KEY (`Perfil\_ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Hitorial`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Hitorial` (

`ID\_HISTORIAL` INT NOT NULL AUTO\_INCREMENT,

`ID\_PERFIL` INT NOT NULL,

`TEXT` TEXT NOT NULL,

`IMAGEN` LONGBLOB NOT NULL,

`LIGA` VARCHAR(45) NOT NULL,

INDEX `fk\_Perfil\_has\_Publicacion\_Perfil1\_idx` (`ID\_PERFIL` ASC),

PRIMARY KEY (`ID\_HISTORIAL`),

CONSTRAINT `fk\_Perfil\_has\_Publicacion\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Publicacion`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Publicacion` (

`ID\_PUBLICACION` INT NOT NULL AUTO\_INCREMENT,

`ID\_HISTORIAL` INT NOT NULL,

`TEXTO` LONGTEXT NOT NULL,

`DIA-HORA` DATETIME NOT NULL,

`FOTO` BLOB NOT NULL,

`ID\_UBICACION` INT NOT NULL,

`ID\_ESTADO` INT NOT NULL,

PRIMARY KEY (`ID\_PUBLICACION`),

INDEX `fk\_Publicacion\_Hitorial1\_idx` (`ID\_HISTORIAL` ASC),

INDEX `fk\_Publicacion\_Ubicacion1\_idx` (`ID\_UBICACION` ASC),

INDEX `fk\_Publicacion\_Estado1\_idx` (`ID\_ESTADO` ASC),

CONSTRAINT `fk\_Publicacion\_Hitorial1`

FOREIGN KEY (`ID\_HISTORIAL`)

REFERENCES `BD\_Facebook`.`Hitorial` (`ID\_HISTORIAL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Publicacion\_Ubicacion1`

FOREIGN KEY (`ID\_UBICACION`)

REFERENCES `BD\_Facebook`.`Ubicacion` (`ID\_UBICACION`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Publicacion\_Estado1`

FOREIGN KEY (`ID\_ESTADO`)

REFERENCES `BD\_Facebook`.`Estado` (`ID\_ESTADO`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Amigo`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Amigo` (

`ID\_PERFIL` INT NOT NULL,

`ID\_AMIGO` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Perfil\_Perfil2\_idx` (`ID\_AMIGO` ASC),

INDEX `fk\_Perfil\_has\_Perfil\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Perfil\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Perfil\_Perfil2`

FOREIGN KEY (`ID\_AMIGO`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Etiquetado`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Etiquetado` (

`ID\_PERFIL` INT NOT NULL,

`ID\_PUBLICACION` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Publicacion\_Publicacion2\_idx` (`ID\_PUBLICACION` ASC),

INDEX `fk\_Perfil\_has\_Publicacion\_Perfil2\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Publicacion\_Perfil2`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Publicacion\_Publicacion2`

FOREIGN KEY (`ID\_PUBLICACION`)

REFERENCES `BD\_Facebook`.`Publicacion` (`ID\_PUBLICACION`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BD\_Facebook`.`Me\_Gusta`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `BD\_Facebook`.`Me\_Gusta` (

`ID\_PERFIL` INT NOT NULL,

`ID\_PUBLICACION` INT NOT NULL,

INDEX `fk\_Perfil\_has\_Publicacion1\_Publicacion1\_idx` (`ID\_PUBLICACION` ASC),

INDEX `fk\_Perfil\_has\_Publicacion1\_Perfil1\_idx` (`ID\_PERFIL` ASC),

CONSTRAINT `fk\_Perfil\_has\_Publicacion1\_Perfil1`

FOREIGN KEY (`ID\_PERFIL`)

REFERENCES `BD\_Facebook`.`Perfil` (`ID\_PERFIL`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Perfil\_has\_Publicacion1\_Publicacion1`

FOREIGN KEY (`ID\_PUBLICACION`)

REFERENCES `BD\_Facebook`.`Publicacion` (`ID\_PUBLICACION`)

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;

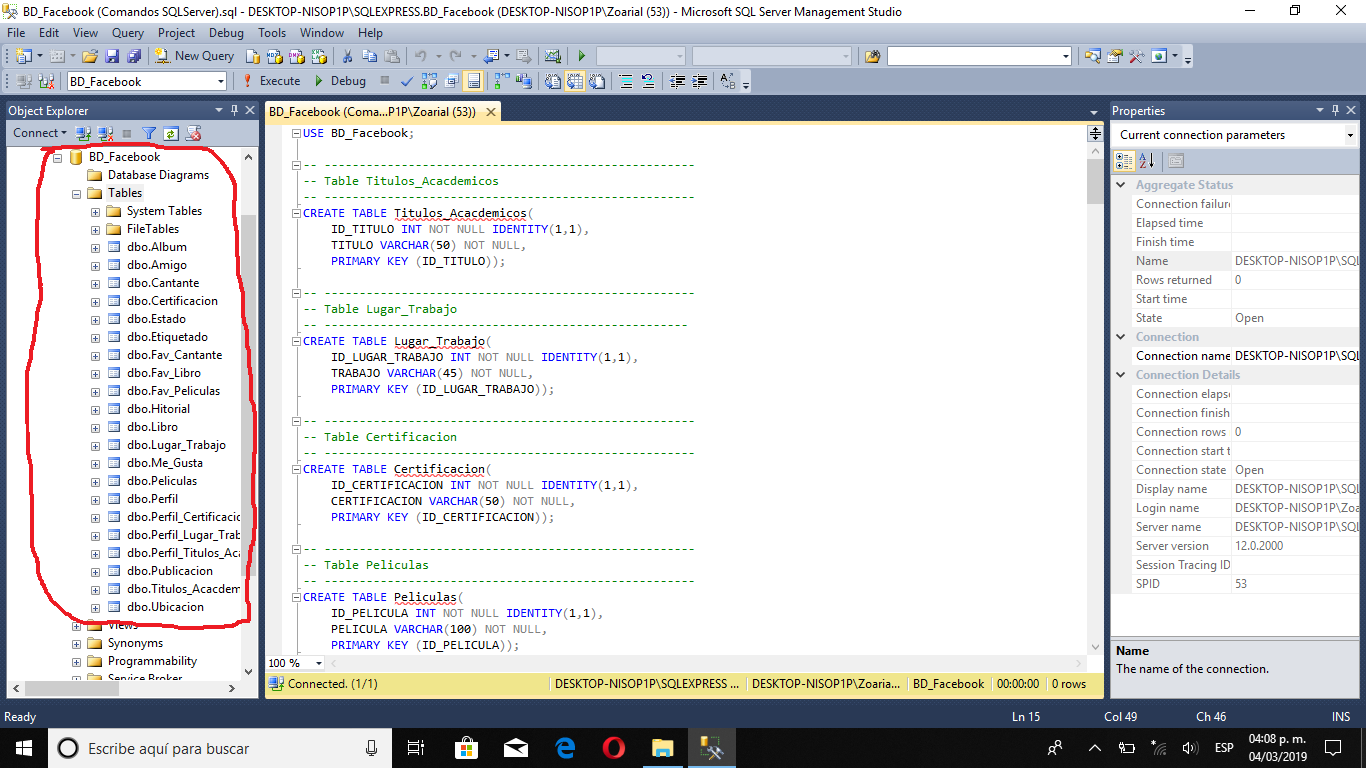
***Índices***

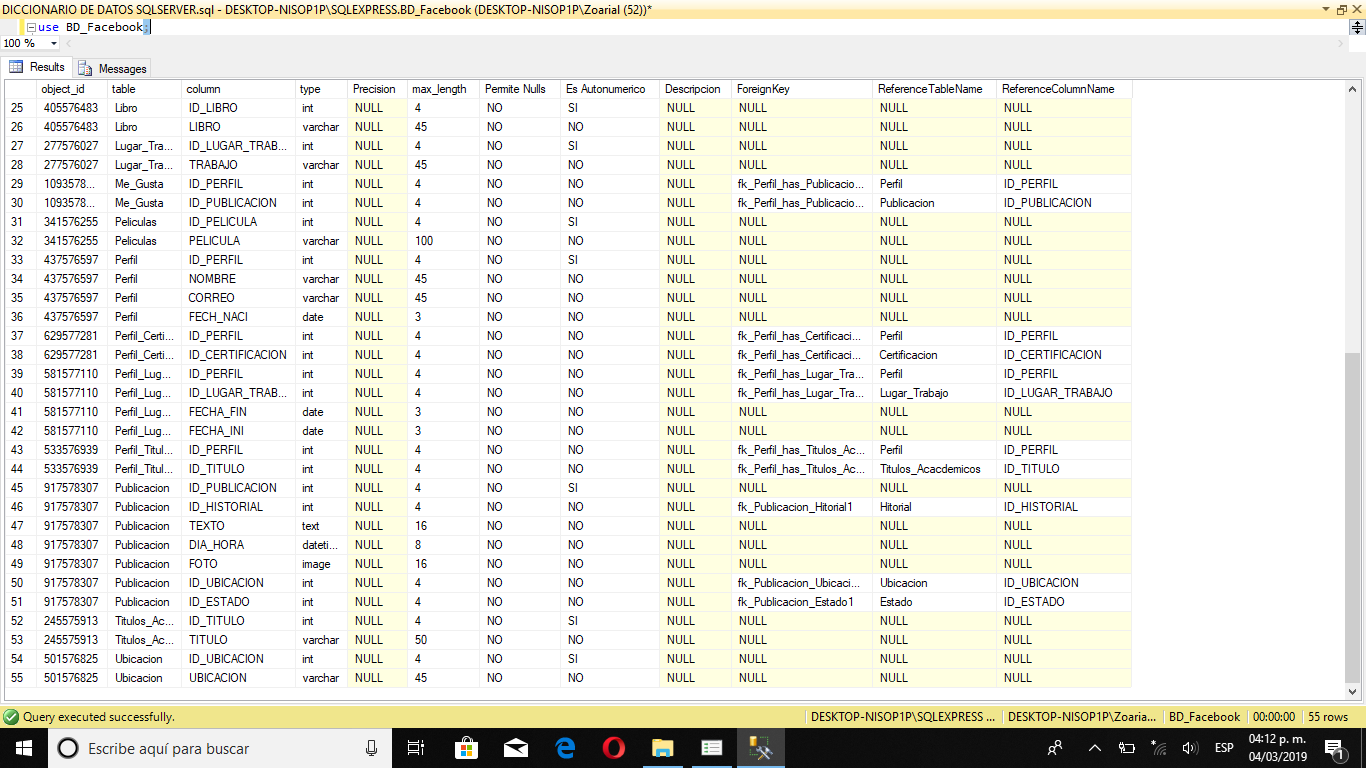
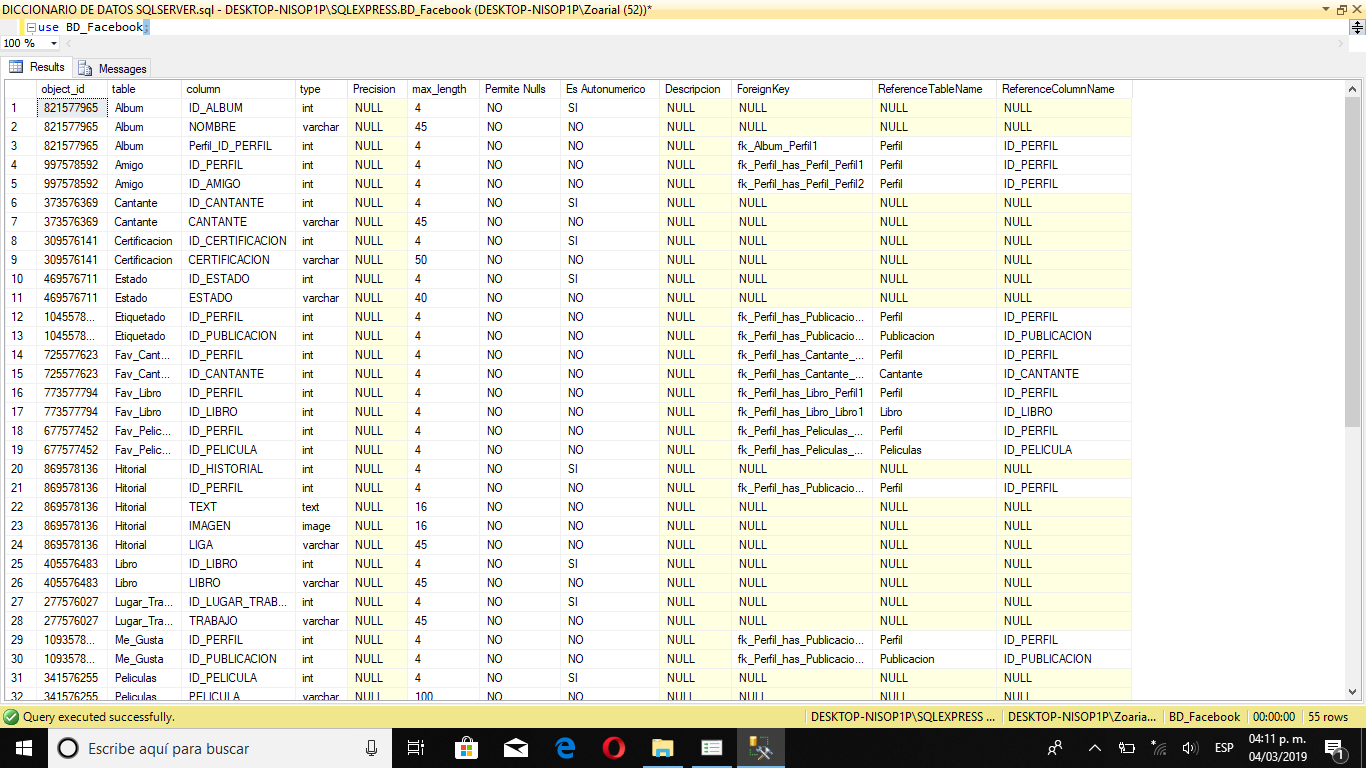
Los campos que pensamos que serían candidatos a crear un índice son:

* ID\_PERFIL además de ID\_AMIGO de la tabla Amigo, ya que por la estructura de la base de datos al momento de hacer consultas sobre la tabla Amigo para ver los amigos de cada perfil, por lo tanto, la creación de un índice nos ayudaría mejorar la velocidad de consultas.
* ID\_Publicacion y ID\_Perfil de la tabla Me\_Gusta, porque para el propósito de la base de datos esta tabla será muy usada y ayudaría a mejorar la velocidad de consulta.
* ID\_Perfil y ID\_Publicacion de la tabla Etiquetado, ya que el uso que se le dará a la base de datos, crear un índice en estos campos nos ayudaría a mejorar la velocidad de consulta.
* TITULO de la tabla Titulos\_Academicos, TRABAJO de la Tabla Lugar\_Trabajo, CERTIFICACION de la tabla Certificacion, PELICULA de la tabla Peliculas, CANTANTE de la tabla Cantante y LIBRO de la tabla Libro , todos estos campos tiene algo en común, eso es que son totalmente o parcialmente directorios, ya que los nuevos registros que tendremos en estas tablas, se agregaran raramente, por lo que al ser casi estáticos un índice nos ayudaría a tener una mejor optimización de la base de datos.

Por último, no descartamos la creación de índices en las llaves primarias y foráneas, pero ya que los índices son creados automáticamente al momento de crear la base de datos no lo mencionamos a fondo.

***Sql Server***

**Base de datos creada**

**Diccionario de datos**