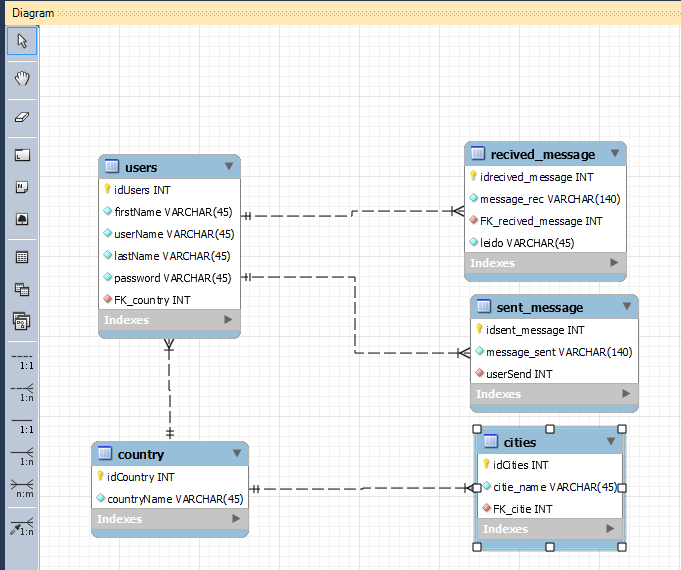
**Trabajo práctico Integrador Base de Datos**

* Estructura de Tablas + Diagrama de entidad relación



**Nota**: Aquí se aplicó al máximo lo que se enseñó en clases sobre la metodología de Normalización de datos.

* Script de creación de la base de datos.

**Nota:** Se subió también la base de datos al repositorio de GIT. Para facilitar su revisión.

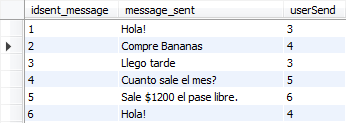
|  |
| --- |
| -- MySQL Script generated by MySQL Workbench  -- Fri Feb 25 00:38:34 2022  -- Model: New Model Version: 1.0  -- MySQL Workbench Forward Engineering  SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;  SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;  SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';  -- -----------------------------------------------------  -- Schema mydb  -- -----------------------------------------------------  -- -----------------------------------------------------  -- Schema final\_bbdd  -- -----------------------------------------------------  -- -----------------------------------------------------  -- Schema final\_bbdd  -- -----------------------------------------------------  CREATE SCHEMA IF NOT EXISTS `final\_bbdd` DEFAULT CHARACTER SET utf8 ;  USE `final\_bbdd` ;  -- -----------------------------------------------------  -- Table `final\_bbdd`.`country`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `final\_bbdd`.`country` (  `idCountry` INT NOT NULL AUTO\_INCREMENT,  `countryName` VARCHAR(45) NOT NULL,  PRIMARY KEY (`idCountry`))  ENGINE = InnoDB  AUTO\_INCREMENT = 9  DEFAULT CHARACTER SET = utf8mb3;  -- -----------------------------------------------------  -- Table `final\_bbdd`.`cities`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `final\_bbdd`.`cities` (  `idCities` INT NOT NULL AUTO\_INCREMENT,  `citie\_name` VARCHAR(45) NOT NULL,  `FK\_citie` INT NOT NULL,  PRIMARY KEY (`idCities`),  INDEX `FK\_citie\_idx` (`FK\_citie` ASC) VISIBLE,  CONSTRAINT `FK\_citie`  FOREIGN KEY (`FK\_citie`)  REFERENCES `final\_bbdd`.`country` (`idCountry`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  AUTO\_INCREMENT = 9  DEFAULT CHARACTER SET = utf8mb3;  -- -----------------------------------------------------  -- Table `final\_bbdd`.`users`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `final\_bbdd`.`users` (  `idUsers` INT NOT NULL AUTO\_INCREMENT,  `firstName` VARCHAR(45) NOT NULL,  `userName` VARCHAR(45) NOT NULL,  `lastName` VARCHAR(45) NOT NULL,  `password` VARCHAR(45) NOT NULL,  `FK\_country` INT NOT NULL,  PRIMARY KEY (`idUsers`),  INDEX `FK\_country\_idx` (`FK\_country` ASC) VISIBLE,  CONSTRAINT `FK\_country`  FOREIGN KEY (`FK\_country`)  REFERENCES `final\_bbdd`.`country` (`idCountry`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  AUTO\_INCREMENT = 8  DEFAULT CHARACTER SET = utf8mb3;  -- -----------------------------------------------------  -- Table `final\_bbdd`.`recived\_message`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `final\_bbdd`.`recived\_message` (  `idrecived\_message` INT NOT NULL AUTO\_INCREMENT,  `message\_rec` VARCHAR(140) NOT NULL,  `FK\_recived\_message` INT NOT NULL,  `leido` VARCHAR(45) NOT NULL,  PRIMARY KEY (`idrecived\_message`),  INDEX `FK\_recived\_message\_idx` (`FK\_recived\_message` ASC) VISIBLE,  CONSTRAINT `FK\_recived\_message`  FOREIGN KEY (`FK\_recived\_message`)  REFERENCES `final\_bbdd`.`users` (`idUsers`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  AUTO\_INCREMENT = 7  DEFAULT CHARACTER SET = utf8mb3;  -- -----------------------------------------------------  -- Table `final\_bbdd`.`sent\_message`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `final\_bbdd`.`sent\_message` (  `idsent\_message` INT NOT NULL AUTO\_INCREMENT,  `message\_sent` VARCHAR(140) NOT NULL,  `userSend` INT NOT NULL,  PRIMARY KEY (`idsent\_message`),  INDEX `FK\_userSend\_idx` (`userSend` ASC) VISIBLE,  CONSTRAINT `FK\_userSend`  FOREIGN KEY (`userSend`)  REFERENCES `final\_bbdd`.`users` (`idUsers`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  AUTO\_INCREMENT = 8  DEFAULT CHARACTER SET = utf8mb3;  USE `final\_bbdd` ;  -- -----------------------------------------------------  -- procedure new\_Usuario  -- -----------------------------------------------------  DELIMITER $$  USE `final\_bbdd`$$  CREATE DEFINER=`root`@`localhost` PROCEDURE `new\_Usuario`(IN VAR1 VARCHAR(45), IN VAR2 VARCHAR(45), IN VAR3 VARCHAR(45),  IN VAR4 VARCHAR(45), IN VAR5 INT)  BEGIN  INSERT INTO users (firstName, userName, lastName, password, FK\_country)  values (VAR1, VAR2, VAR3, VAR4, VAR5);  END$$  DELIMITER ;  -- -----------------------------------------------------  -- procedure new\_sentMessage  -- -----------------------------------------------------  DELIMITER $$  USE `final\_bbdd`$$  CREATE DEFINER=`root`@`localhost` PROCEDURE `new\_sentMessage`(IN VAR1 VARCHAR(140), IN VAR2 INT)  BEGIN  INSERT INTO sent\_message (message\_sent, userSend)  values (VAR1, VAR2);  END$$  DELIMITER ;  SET SQL\_MODE=@OLD\_SQL\_MODE;  SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;  SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS; |

**Tablas:**

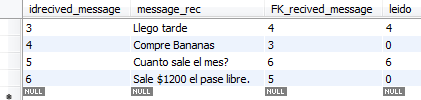
1. **Usuarios**

****

1. **Send\_message**

****

1. **recived\_message**

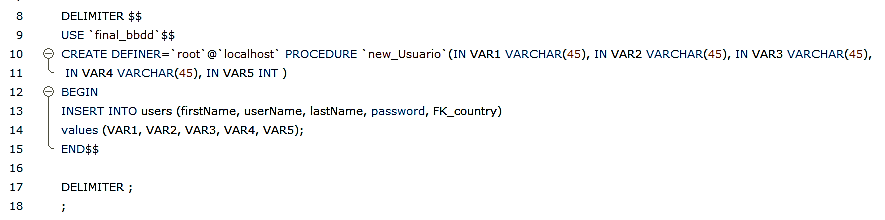
****

1. **Country**

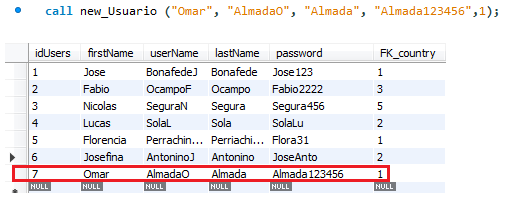
****

**Store Procedure – Workbench**

1. Registro de usuario



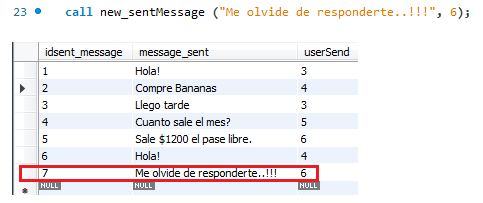
Resultado:



1. Envío de mensaje

****

Resultado:



**Consultas**

1. Cantidad de usuarios por país.

|  |
| --- |
| SELECT countryName, COUNT(FK\_country)  FROM users AS u  JOIN country AS c  ON u.FK\_country = c.idCountry  GROUP BY c.countryName; |
| Resultado: |

1. Cantidad de mensajes por usuario.

|  |
| --- |
| SELECT userName, COUNT(userSend)  FROM sent\_message AS u  JOIN users AS c  ON u.userSend = c.idUsers  GROUP BY c.userName; |
| Resultado: |

1. Cantidad de mensajes leídos por usuario

|  |
| --- |
| SELECT userName, COUNT(leido)  FROM recived\_message AS u  JOIN users AS c  ON u.leido = c.idUsers  GROUP BY c.userName ; |
| Resultado: |