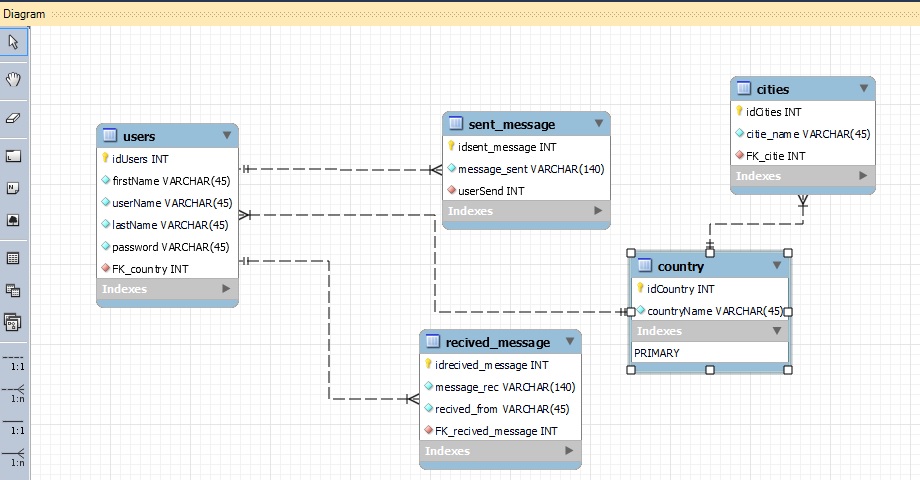
* Estructura de Tablas + Diagrama de entidad relación



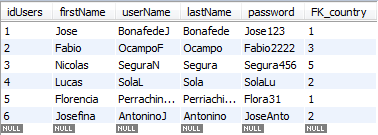
**Nota**: Aquí se aplicó al máximo lo que se explicó en clases la metodología de Normalización de datos

* Script de creación de la base de datos.

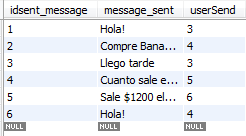
|  |
| --- |
| * -- MySQL Workbench Forward Engineering * SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0; * SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0; * SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION'; * -- ----------------------------------------------------- * -- Schema 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; * -- ----------------------------------------------------- * -- 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; * -- ----------------------------------------------------- * -- 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, * `recived\_from` VARCHAR(45) NOT NULL, * `FK\_recived\_message` INT 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; * -- ----------------------------------------------------- * -- 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; * -- ----------------------------------------------------- * -- 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; * SET SQL\_MODE=@OLD\_SQL\_MODE; * SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS; * SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS; |

**Tablas:**

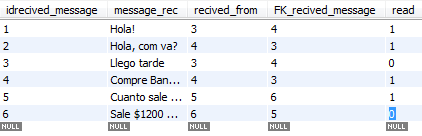
1. **Usuarios**

****

1. **Send\_message**

****

1. **Send\_message**

****

1. **Country**

****

**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