



	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU010104373 9
Versión: 2.0		Fecha : 25/01/20

Se debe entregar un enlace a github al producto desarrollado en cada una de las fases del proyecto, y un archivo zip con el proyecto git.

El Readme de Github debe incluir una descripción GENERACIÓN DE CÓDIGO DE COMPONENTES Y PROCEDIMIENTOS (CSI-2)

NOTA : SE ADJUNTAN LOS FICHEROS CORRESPONDIENTES EN PDF POR SI NO SE VE CLARO CON ESTA PLANTILLA.

CÓDIGO DE COMPONENTES
<p>Script SQL de generación de tablas (tanto para BDR como para BDROO) y objetos (BDROO)</p> <pre>-- MySQL Script generated by MySQL Workbench -- Fri Jan 24 23:30:18 2020 -- 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 mydb ----- CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ; USE `mydb` ; ----- -- Table `mydb`.`Direccion` ----- CREATE TABLE IF NOT EXISTS `mydb`.`Direccion` (`id` INT NOT NULL, `ciudad` VARCHAR(45) NULL, `calle` VARCHAR(45) NULL, `numero` INT NULL, `codigo_postal` CHAR(5) NULL, PRIMARY KEY (`id`)) ENGINE = InnoDB;</pre>

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0		Fecha : 25/01/20

-- Table `mydb`.`Usuario`

CREATE TABLE IF NOT EXISTS `mydb`.`Usuario` (
 `id` INT NOT NULL,
 `nombre` VARCHAR(45) NULL,
 `apellido` VARCHAR(45) NULL,
 `numero_copias_hechas` INT NOT NULL,
 `DNI` VARCHAR(10) NULL,
 `direccion_id` INT NULL,
 PRIMARY KEY (`id`),
 INDEX `direccion_idx` (`direccion_id` ASC) VISIBLE,
 CONSTRAINT `direccion`
 FOREIGN KEY (`direccion_id`)
 REFERENCES `mydb`.`Direccion` (`id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- Table `mydb`.`Categoria`

CREATE TABLE IF NOT EXISTS `mydb`.`Categoria` (
 `id` INT NOT NULL,
 `nombre` VARCHAR(25) NOT NULL,
 PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- Table `mydb`.`Autor`

CREATE TABLE IF NOT EXISTS `mydb`.`Autor` (
 `id` INT NOT NULL,
 `name` VARCHAR(45) NOT NULL,
 `apellido` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- Table `mydb`.`Recurso`

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0		Fecha : 25/01/20


```

-----
CREATE TABLE IF NOT EXISTS `mydb`.`Recurso` (
  `id` INT NOT NULL,
  `tipo` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-----
-- Table `mydb`.`Libro`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Libro` (
  `recurso_id` INT NOT NULL,
  `ISBN` CHAR(13) NOT NULL,
  `titulo` VARCHAR(45) NULL,
  `numero_de_paginas` INT NULL,
  `area` VARCHAR(10) NULL,
  `estante` INT NULL,
  `categoria_id` INT NULL,
  `tipo_casa_o_sala` CHAR(1) NOT NULL,
  `autor_id` INT NULL,
  INDEX `fk_book_book_category1_idx` (`categoria_id` ASC) VISIBLE,
  INDEX `autor_idx` (`autor_id` ASC) VISIBLE,
  PRIMARY KEY (`ISBN`),
  CONSTRAINT `categoria`
    FOREIGN KEY (`categoria_id`)
      REFERENCES `mydb`.`Categoria` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `autor`
    FOREIGN KEY (`autor_id`)
      REFERENCES `mydb`.`Autor` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `recurso_padre`
    FOREIGN KEY (`recurso_id`)
      REFERENCES `mydb`.`Recurso` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`Empleado`
-----

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0		Fecha : 25/01/20


```
CREATE TABLE IF NOT EXISTS `mydb`.`Empleado` (
  `seguridad_social` VARCHAR(20) NOT NULL,
  `turno` VARCHAR(45) NULL,
  `horario` VARCHAR(45) NULL,
  `salario` INT NULL,
  `direccion_id` INT NOT NULL,
  PRIMARY KEY (`seguridad_social`),
  INDEX `fk_Empleado_Direccion1_idx` (`direccion_id` ASC) VISIBLE,
  CONSTRAINT `fk_Empleado_Direccion1`
    FOREIGN KEY (`direccion_id`)
      REFERENCES `mydb`.`Direccion` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Revista`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Revista` (
  `recurso_id` INT NOT NULL,
  `titulo` VARCHAR(45) NOT NULL,
  `fotocopias_realizadas` TINYINT(1) NOT NULL,
  PRIMARY KEY (`titulo`),
  CONSTRAINT `recurso_padre3`
    FOREIGN KEY (`recurso_id`)
      REFERENCES `mydb`.`Recurso` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Portatil`
```


```
CREATE TABLE IF NOT EXISTS `mydb`.`Portatil` (
  `recurso_id` INT NOT NULL,
  `marca` VARCHAR(45) NOT NULL,
  `modelo` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`modelo`, `marca`),
  CONSTRAINT `recurso_padre2`
    FOREIGN KEY (`recurso_id`)
      REFERENCES `mydb`.`Recurso` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0		Fecha : 25/01/20


ENGINE = InnoDB;

-- Table `mydb`.`Prestamo`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Prestamo` (
  `fecha_inicio` DATE NOT NULL,
  `fecha_fin` DATE NOT NULL,
  `usuario_id` INT NOT NULL,
  `empleado_seguridad_social` VARCHAR(20) NOT NULL,
  `casa_o_sala` VARCHAR(4) NOT NULL,
  `Libro_ISBN` CHAR(13) NULL,
  `Revista_titulo` VARCHAR(45) NULL,
  `Portatil_modelo` VARCHAR(45) NULL,
  `Portatil_marca` VARCHAR(45) NULL,
  INDEX `fk_book_borrow_service_student1_idx` (`usuario_id` ASC) VISIBLE,
  INDEX `fk_book_reservation_library_member1_idx` (`empleado_seguridad_social` ASC) VISIBLE,
  PRIMARY KEY (`empleado_seguridad_social`, `usuario_id`, `fecha_inicio`),
  INDEX `fk_Prestamo_Libro1_idx` (`Libro_ISBN` ASC) VISIBLE,
  INDEX `fk_Prestamo_Revista1_idx` (`Revista_titulo` ASC) VISIBLE,
  INDEX `fk_Prestamo_Portatil1_idx` (`Portatil_modelo` ASC, `Portatil_marca` ASC) VISIBLE,
  CONSTRAINT `usuario`
    FOREIGN KEY (`usuario_id`)
    REFERENCES `mydb`.`Usuario` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `empleado`
    FOREIGN KEY (`empleado_seguridad_social`)
    REFERENCES `mydb`.`Empleado` (`seguridad_social`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Prestamo_Libro1`
    FOREIGN KEY (`Libro_ISBN`)
    REFERENCES `mydb`.`Libro` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Prestamo_Revista1`
    FOREIGN KEY (`Revista_titulo`)
    REFERENCES `mydb`.`Revista` (`titulo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Prestamo_Portatil1`
    FOREIGN KEY (`Portatil_modelo`, `Portatil_marca`)
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0		Fecha : 25/01/20


<p>REFERENCES `mydb`.`Portatil` (`modelo` , `marca`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;</p> <p>-----</p> <p>-- Table `mydb`.`Sala`</p> <p>-----</p> <p>CREATE TABLE IF NOT EXISTS `mydb`.`Sala` (`id` INT NOT NULL, `capacidad` INT NULL, PRIMARY KEY (`id`)) ENGINE = InnoDB;</p> <p>-----</p> <p>-- Table `mydb`.`reservacion_de_sala`</p> <p>-----</p> <p>CREATE TABLE IF NOT EXISTS `mydb`.`reservacion_de_sala` (`sala_id` INT NOT NULL, `fecha` DATE NOT NULL, `hora_inicio` TIME NOT NULL, `hora_fin` TIME NOT NULL, `empleado_seguridad_social` VARCHAR(20) NOT NULL, `usuario_id` INT NOT NULL, INDEX `fk_reading_room_reservation_library_member1_idx` (`empleado_seguridad_social` ASC) VISIBLE, INDEX `fk_reading_room_reservation_student1_idx` (`usuario_id` ASC) VISIBLE, PRIMARY KEY (`sala_id`, `hora_inicio`, `empleado_seguridad_social`, `usuario_id`, `fecha`), CONSTRAINT `empleado2` FOREIGN KEY (`empleado_seguridad_social`) REFERENCES `mydb`.`Empleado` (`seguridad_social`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `usuario2` FOREIGN KEY (`usuario_id`) REFERENCES `mydb`.`Usuario` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `sala` FOREIGN KEY (`sala_id`) REFERENCES `mydb`.`Sala` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION)</p>

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0	Tiempo invertido:20 h	Fecha : 25/01/20

```
ENGINE = InnoDB;

-----
-- Table `mydb`.`Delito`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Delito` (
  `fecha` DATE NOT NULL,
  `usuario_id` INT NOT NULL,
  `empleado_seguridad_social` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`fecha`, `usuario_id`),
  INDEX `fk_deadline_infraction_student1_idx` (`usuario_id` ASC) VISIBLE,
  INDEX `fk_deadline_infraction_library_member1_idx` (`empleado_seguridad_social` ASC) VISIBLE,
  CONSTRAINT `usuario3`
    FOREIGN KEY (`usuario_id`)
    REFERENCES `mydb`.`Usuario` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `empleado3`
    FOREIGN KEY (`empleado_seguridad_social`)
    REFERENCES `mydb`.`Empleado` (`seguridad_social`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`Charla`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Charla` (
  `titulo` VARCHAR(45) NOT NULL,
  `usuario_id` INT NULL,
  `categoria_id` INT NULL,
  PRIMARY KEY (`titulo`),
  INDEX `usuario_idx` (`usuario_id` ASC) VISIBLE,
  INDEX `categoria_idx` (`categoria_id` ASC) VISIBLE,
  CONSTRAINT `usuario4`
    FOREIGN KEY (`usuario_id`)
    REFERENCES `mydb`.`Usuario` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `categoria3`
    FOREIGN KEY (`categoria_id`)
    REFERENCES `mydb`.`Categoria` (`id`))
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca	Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0	Tiempo invertido:20 h	Fecha : 25/01/20

```

ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `mydb`.`Reaccion`
-----


CREATE TABLE IF NOT EXISTS `mydb`.`Reaccion` (
  `id` INT NOT NULL,
  `delito_fecha` DATE NOT NULL,
  `usuario_id` INT NOT NULL,
  `mensaje` VARCHAR(45) NULL,
  `fecha` DATE NULL,
  PRIMARY KEY (`id`, `delito_fecha`, `usuario_id`),
  INDEX `delito_idx` (`delito_fecha` ASC, `usuario_id` ASC) VISIBLE,
  CONSTRAINT `delito`
    FOREIGN KEY (`delito_fecha`, `usuario_id`)
      REFERENCES `mydb`.`Delito` (`fecha`, `usuario_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `mydb`.`Prestamo_portatil`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`Prestamo_portatil` (
  `fecha_inicio` DATE NOT NULL,
  `fecha_fin` DATE NOT NULL,
  `usuario_id` INT NOT NULL,
  `empleado_seguridad_social` VARCHAR(20) NOT NULL,
  `portatil_id` INT NOT NULL,
  INDEX `fk_book_borrow_service_student1_idx` (`usuario_id` ASC) VISIBLE,
  INDEX `fk_book_reservation_library_member1_idx` (`empleado_seguridad_social` ASC) VISIBLE,
  PRIMARY KEY (`empleado_seguridad_social`, `usuario_id`, `fecha_inicio`, `portatil_id`),
  INDEX `portatil_idx` (`portatil_id` ASC) VISIBLE,
  CONSTRAINT `usuario0`
    FOREIGN KEY (`usuario_id`)
      REFERENCES `mydb`.`Usuario` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `empleado0`
    FOREIGN KEY (`empleado_seguridad_social`)

```


 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI) PROYECTO: Biblioteca	BASES DE DATOS Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0	Tiempo invertido:20 h	Fecha : 25/01/20

```


REFERENCES `mydb`.`Empleado` (`seguridad_social`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `portatil`
FOREIGN KEY (`portatil_id`)
REFERENCES `mydb`.`Portatil` (`recurso_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`Coleccion`
-----


CREATE TABLE IF NOT EXISTS `mydb`.`Coleccion` (
  `id` INT NOT NULL,
  `usuario_id` INT NOT NULL,
  `coleccion` LONGBLOB NOT NULL,
  PRIMARY KEY (`id`, `usuario_id`),
  INDEX `usuario_idx` (`usuario_id` ASC) VISIBLE,
  CONSTRAINT `usuario5`
    FOREIGN KEY (`usuario_id`)
    REFERENCES `mydb`.`Usuario` (`id`)
    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;

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

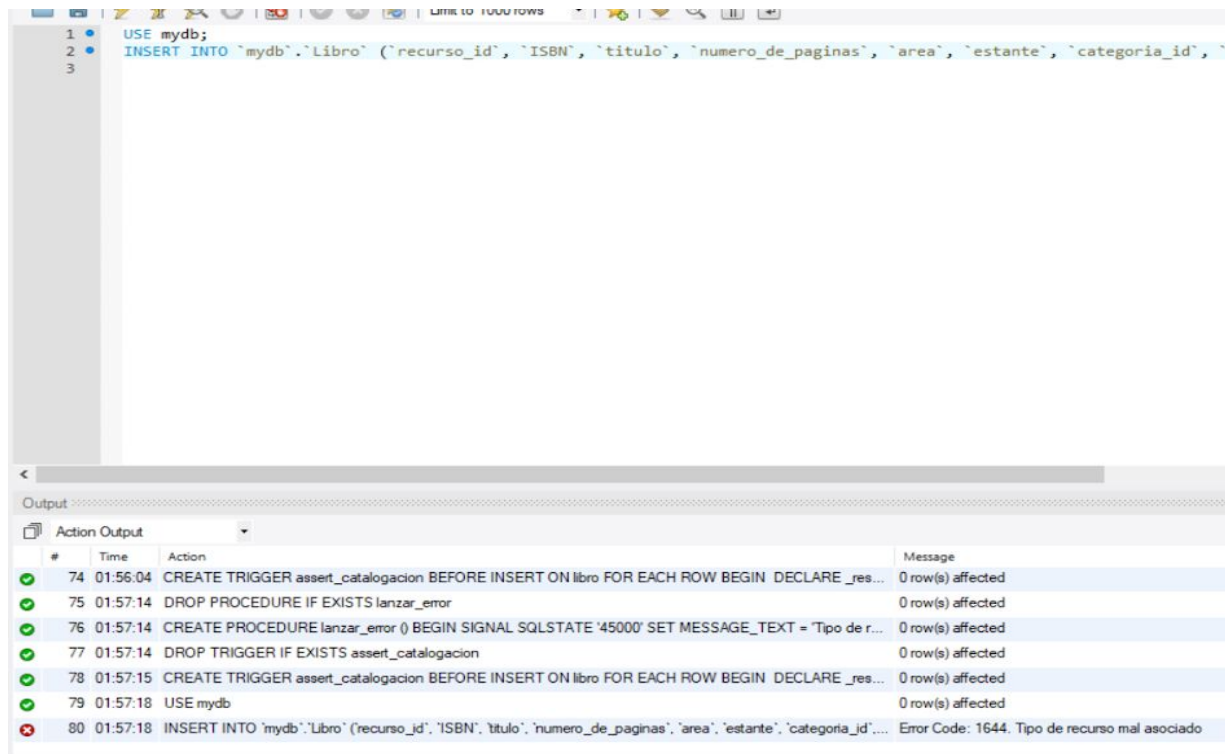
CÓDIGO DE PROCEDIMIENTOS DE OPERACIÓN Y SEGURIDAD
<p>Código SQL de los asertos y disparadores presentes en la base de datos</p> <p>Pruebas de triggers implementados:</p> <p>1.- Todos los recursos se han de catalogar en uno de los tres tipos para ello : Libro, revista o portatil. Se crea un trigger para comprobar que se asocia un recurso dentro del dominio correspondiente : libro - 1, revista - 2 o portatil - 3. Ejemplo de trigger para libro:</p> <pre> USE mydb: delimiter // DROP PROCEDURE IF EXISTS lanzar_error // CREATE PROCEDURE lanzar_error () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tipo de recurso mal asociado'; END// delimiter ; DELIMITER \$\$ DROP TRIGGER IF EXISTS assert_catalogacion \$\$ CREATE TRIGGER assert_catalogacion BEFORE INSERT ON libro FOR EACH ROW BEGIN DECLARE _result,_ISBN BOOL DEFAULT FALSE; IF (NEW.recurso_id>3) THEN SET _result=TRUE; END IF; IF (_result) THEN CALL lanzar_error(); END IF; IF (NEW.ISBN!=NULL AND NEW.recurso_id!=1) THEN SET _ISBN=TRUE; END IF; IF (_ISBN) THEN CALL lanzar_error(); </pre>

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

END IF;

END \$\$

DELIMITER ;



The screenshot displays a SQL IDE interface. The top pane shows the following SQL code:


```

1  USE mydb;
2  INSERT INTO `mydb`.`Libro` (`recurso_id`, `ISBN`, `titulo`, `numero_de_paginas`, `area`, `estante`, `categoria_id`, `
3

```

The bottom pane shows the 'Output' window with the 'Action Output' tab selected. It contains a table of execution results:

#	Time	Action	Message
74	01:56:04	CREATE TRIGGER assert_catalogacion BEFORE INSERT ON libro FOR EACH ROW BEGIN DECLARE _res...	0 row(s) affected
75	01:57:14	DROP PROCEDURE IF EXISTS lanzar_error	0 row(s) affected
76	01:57:14	CREATE PROCEDURE lanzar_error () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tipo de r...	0 row(s) affected
77	01:57:14	DROP TRIGGER IF EXISTS assert_catalogacion	0 row(s) affected
78	01:57:15	CREATE TRIGGER assert_catalogacion BEFORE INSERT ON libro FOR EACH ROW BEGIN DECLARE _res...	0 row(s) affected
79	01:57:18	USE mydb	0 row(s) affected
80	01:57:18	INSERT INTO `mydb`.`Libro` (`recurso_id`, `ISBN`, `titulo`, `numero_de_paginas`, `area`, `estante`, `categoria_id`,	Error Code: 1644. Tipo de recurso mal asociado

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU010104373 9
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

2.- Si el usuario devuelve un libro más tarde de los días que tiene permitido, se envía de inmediato un recordatorio como primer aviso; (1 día mas tarde por poner un ejemplo)

USE mydb:

delimiter //

DROP PROCEDURE IF EXISTS enviar_aviso //

CREATE PROCEDURE enviar_aviso (IN candidato INT)

BEGIN

INSERT INTO `mydb`.`Reaccion` (`id`, `delito_fecha`, `usuario_id`, `mensaje`, `fecha`) VALUES (10, '2019-05-06', candidato, 'Enviado aviso', '2019-05-08');

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El aviso ha sido enviado al usuario correspondiente.';

END//

delimiter ;

DELIMITER \$\$

DROP TRIGGER IF EXISTS enviar_aviso_trigger \$\$

CREATE TRIGGER enviar_aviso_trigger

BEFORE UPDATE ON prestamo FOR EACH ROW


BEGIN

DECLARE _result BOOL DEFAULT FALSE;

if NEW.fecha_fin>DATE_ADD(NEW.fecha_inicio, INTERVAL 1 DAY) THEN
SET _RESULT=TRUE;

END IF;

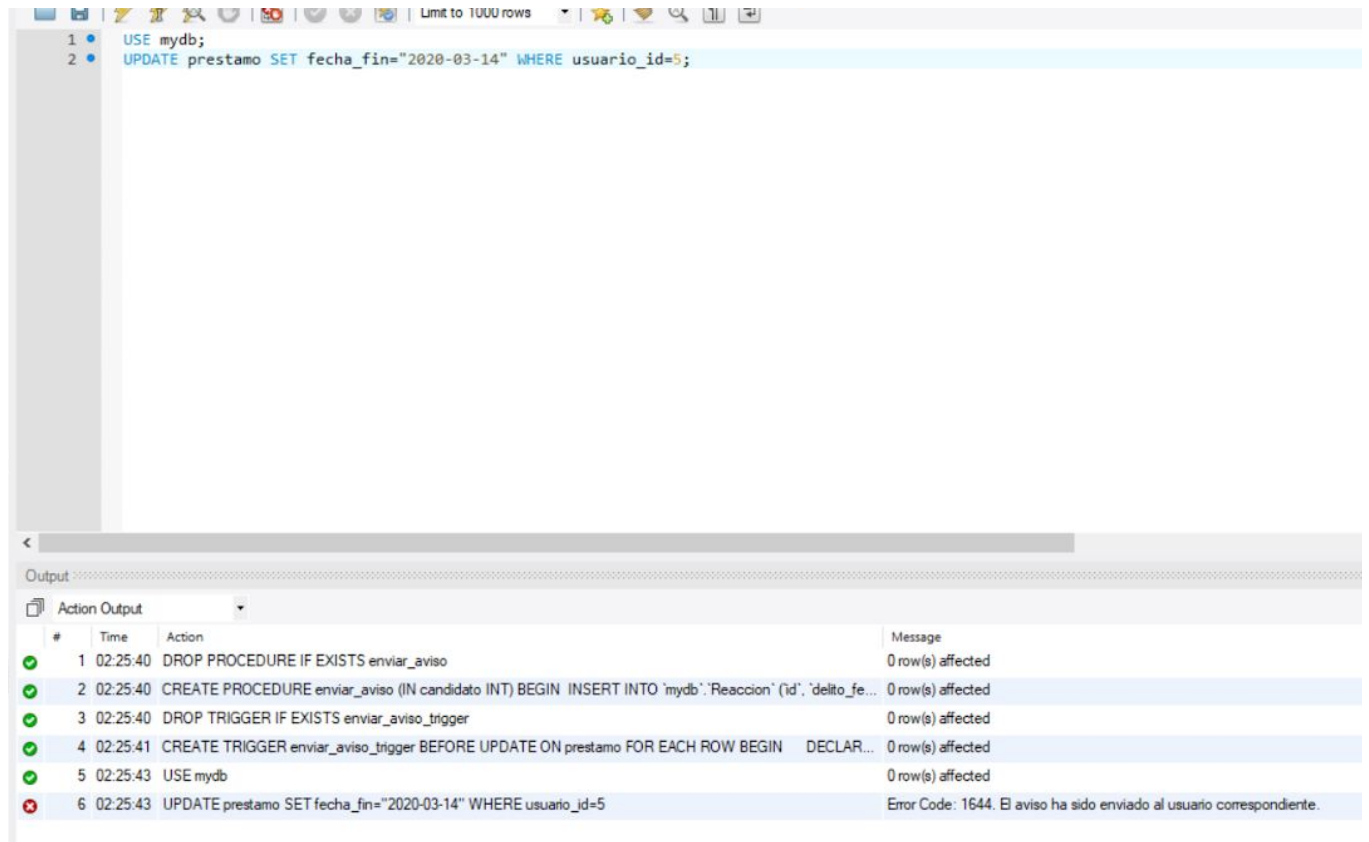
if (_result) THEN
CALL enviar_aviso(NEW.usuario_id);

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU010104373 9
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

END IF;

END \$\$

DELIMITER ;



The screenshot shows a MySQL command-line interface with the following SQL commands and their output:


```

1 • USE mydb;
2 • UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5;

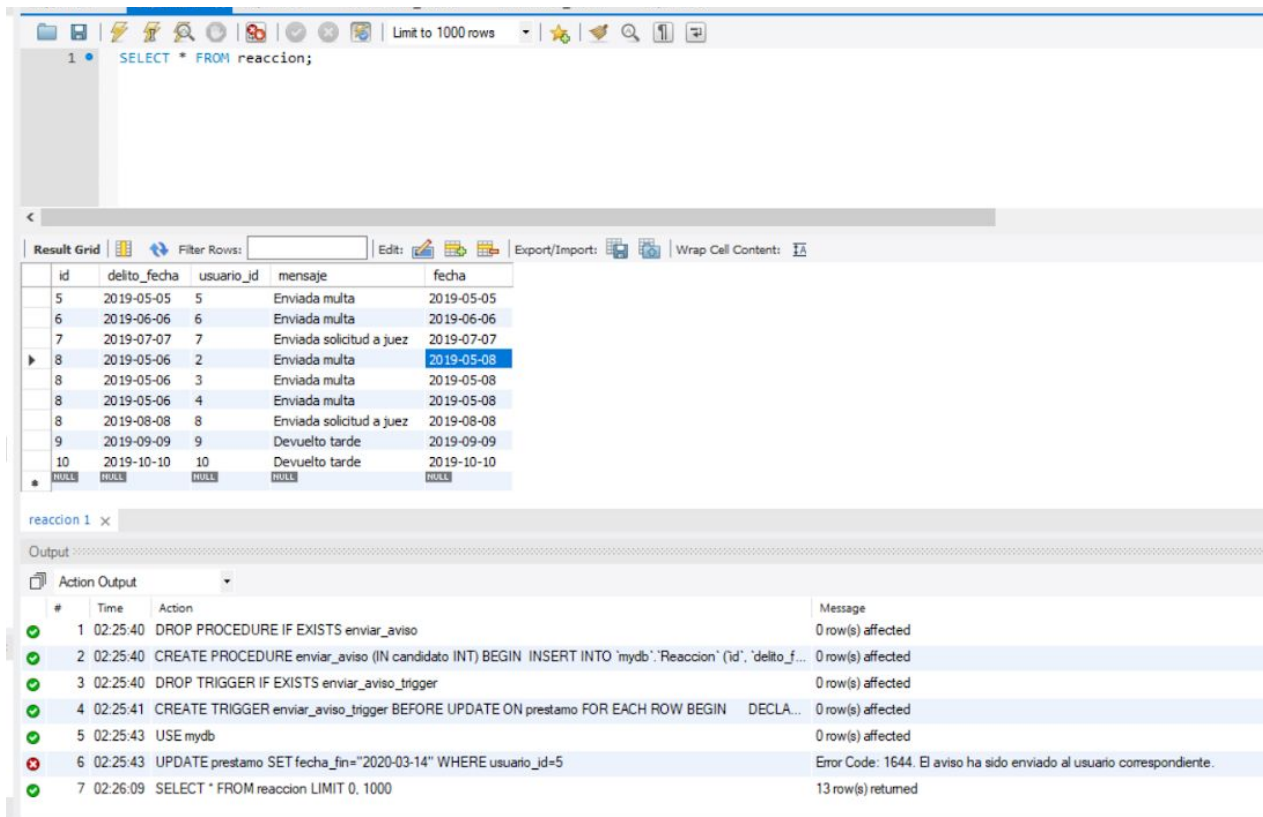
```

The output window shows the following results:

#	Time	Action	Message
1	02:25:40	DROP PROCEDURE IF EXISTS enviar_aviso	0 row(s) affected
2	02:25:40	CREATE PROCEDURE enviar_aviso (IN candidato INT) BEGIN INSERT INTO `mydb`.`Reaccion` (`id`, `delito_fe...	0 row(s) affected
3	02:25:40	DROP TRIGGER IF EXISTS enviar_aviso_trigger	0 row(s) affected
4	02:25:41	CREATE TRIGGER enviar_aviso_trigger BEFORE UPDATE ON prestamo FOR EACH ROW BEGIN DECLAR...	0 row(s) affected
5	02:25:43	USE mydb	0 row(s) affected
6	02:25:43	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5	Error Code: 1644. El aviso ha sido enviado al usuario correspondiente.

 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI) PROYECTO: Biblioteca	BASES DE DATOS Generación de código
	Autor:Alu0101043739	ALU0101043739
Versión: 2.0		Tiempo invertido:20 h
		Fecha : 25/01/20

Vemos que efectivamente se ha creado una alerta con esa fecha:



The screenshot shows a database management interface. At the top, a query is entered: `SELECT * FROM reaccion;`. Below the query, a table displays the results of the query. The table has columns: `id`, `delito_fecha`, `usuario_id`, `mensaje`, and `fecha`. The results show several rows of data, with the row `(8, '2019-05-06', 2, 'Enviada multa', '2019-05-08')` highlighted in blue. Below the table, an 'Output' section shows the execution log of the query. The log includes the following entries:

#	Time	Action	Message
1	02:25:40	DROP PROCEDURE IF EXISTS enviar_aviso	0 row(s) affected
2	02:25:40	CREATE PROCEDURE enviar_aviso (IN candidato INT) BEGIN INSERT INTO `mydb`.`Reaccion` (`id`, `delito_f...	0 row(s) affected
3	02:25:40	DROP TRIGGER IF EXISTS enviar_aviso_trigger	0 row(s) affected
4	02:25:41	CREATE TRIGGER enviar_aviso_trigger BEFORE UPDATE ON prestamo FOR EACH ROW BEGIN DECLA...	0 row(s) affected
5	02:25:43	USE mydb	0 row(s) affected
6	02:25:43	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5	Error Code: 1644. El aviso ha sido enviado al usuario correspondiente.
7	02:26:09	SELECT * FROM reaccion LIMIT 0, 1000	13 row(s) returned

3. - Una multa financiera como segundo aviso; (2 días mas tarde)

USE mydb:

```

delimiter //
DROP PROCEDURE IF EXISTS enviar_multa //
CREATE PROCEDURE enviar_multa (IN candidato INT)
BEGIN

INSERT INTO `mydb`.`Reaccion` (`id`, `delito_fecha`, `usuario_id`, `mensaje`, `fecha`) VALUES (8,
'2019-05-06', candidato, 'Enviada multa', '2019-05-08');


SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La multa ha sido enviada al usuario correspondiente.';

END//

delimiter ;

DELIMITER $$
DROP TRIGGER IF EXISTS enviar_multa_trigger $$
CREATE TRIGGER enviar_multa_trigger
BEFORE UPDATE ON prestamo FOR EACH ROW
BEGIN

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

```

DECLARE _result BOOL DEFAULT FALSE;

if NEW.fecha_fin>DATE_ADD(NEW.fecha_inicio, INTERVAL 2 DAY) THEN
    SET _RESULT=TRUE;

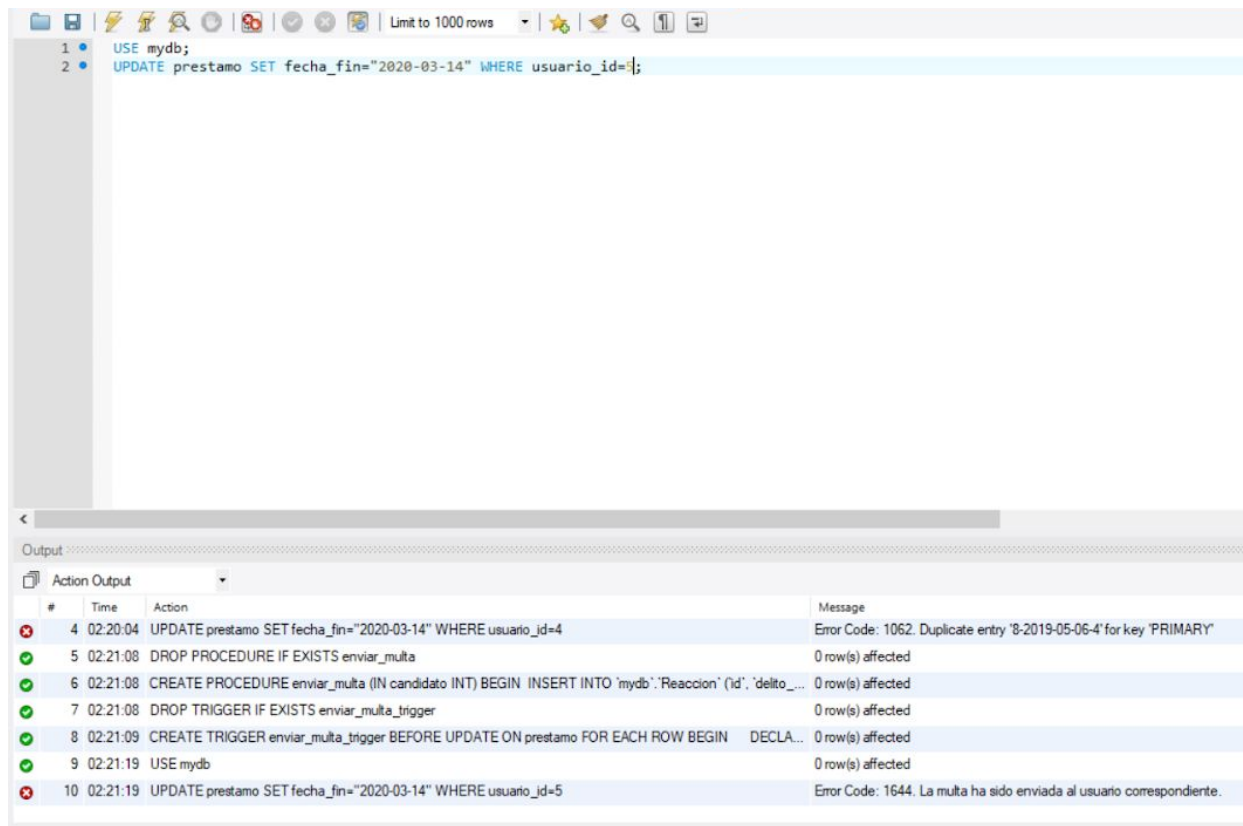
END IF;

if (_result) THEN
    CALL enviar_multa(NEW.usuario_id);

END IF;

END $$
DELIMITER ;


```



The screenshot shows a MySQL command line interface with the following commands and results:

#	Time	Action	Message
1	02:20:04	USE mydb;	
2	02:21:08	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=4;	Error Code: 1062. Duplicate entry '8-2019-05-06-4' for key 'PRIMARY'
3	02:21:08	DROP PROCEDURE IF EXISTS enviar_multa	0 row(s) affected
4	02:21:08	CREATE PROCEDURE enviar_multa (IN candidato INT) BEGIN INSERT INTO 'mydb'. 'Reaccion' ('id', 'delito', 'fecha_fin', 'usuario_id') VALUES (candidato, 'Falta de pago de multa', '2020-03-14', candidato);	0 row(s) affected
5	02:21:08	DROP TRIGGER IF EXISTS enviar_multa_trigger	0 row(s) affected
6	02:21:09	CREATE TRIGGER enviar_multa_trigger BEFORE UPDATE ON prestamo FOR EACH ROW BEGIN DECLARE _result BOOL; IF NEW.fecha_fin > DATE_ADD(NEW.fecha_inicio, INTERVAL 2 DAY) THEN SET _result = TRUE; END IF; IF (_result) THEN CALL enviar_multa(NEW.usuario_id); END IF; END	0 row(s) affected
7	02:21:19	USE mydb	0 row(s) affected
8	02:21:19	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5	Error Code: 1644. La multa ha sido enviada al usuario correspondiente.

Observamos el error #4 como demostración , ya que al darle dos veces se había creado ya la primera vez el mensaje y entonces dice que está duplicado.

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU010104373 9
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

4.- Parte de la literatura (de los libros) de la biblioteca está permitida solo para las salas que se reservan (salas de lectura), y la literatura que tiene este código especial se considerará por separado.

USE mydb:

delimiter //

DROP PROCEDURE IF EXISTS enviar_mensaje //

CREATE PROCEDURE enviar_mensaje ()

BEGIN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede llevar este libro a su casa. Lo sentimos';

END//

delimiter ;

DELIMITER \$\$

DROP TRIGGER IF EXISTS libro_casa \$\$

CREATE TRIGGER libro_casa

BEFORE INSERT ON prestamo FOR EACH ROW

BEGIN

DECLARE _result BOOL DEFAULT FALSE;


if ((SELECT tipo_casa_o_sala FROM libro WHERE NEW.Libro_ISBN = libro.ISBN) = 's') THEN

SET _RESULT=TRUE;

END IF;

if (_result) THEN

CALL enviar_mensaje();

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

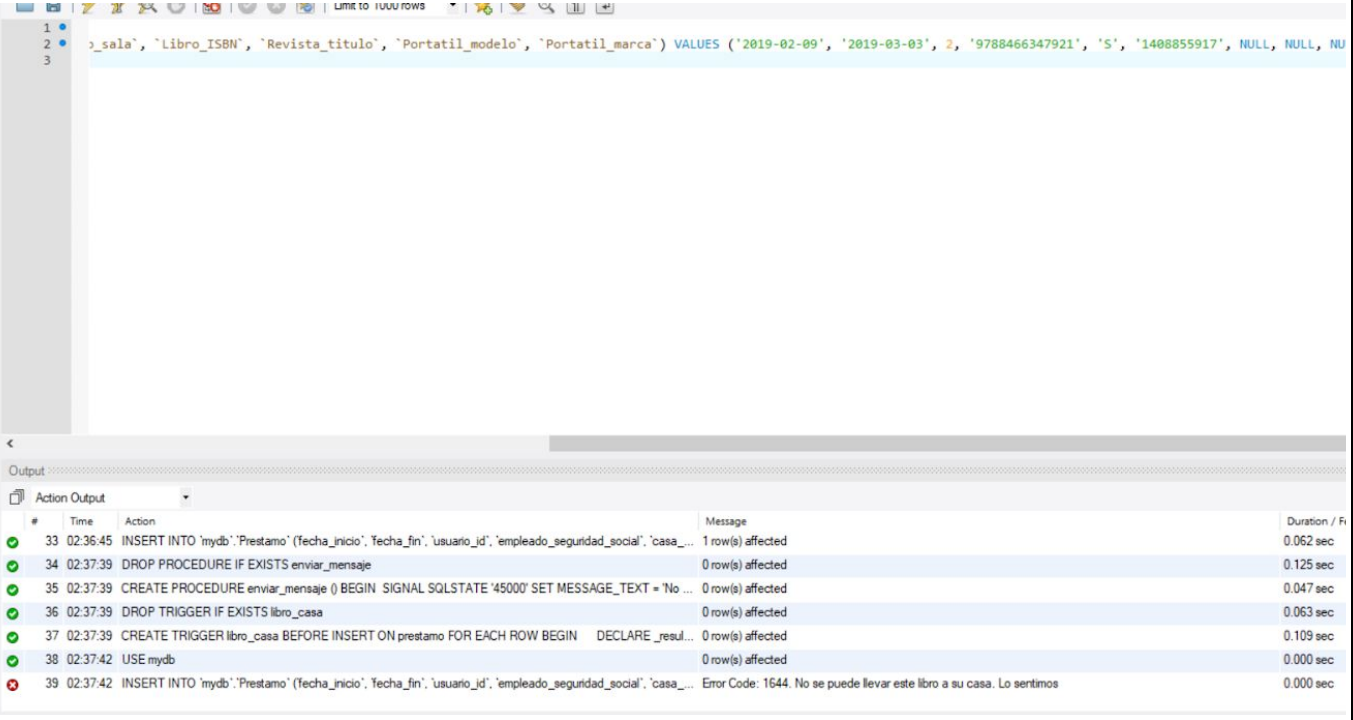
```

END IF;


END $$

DELIMITER ;

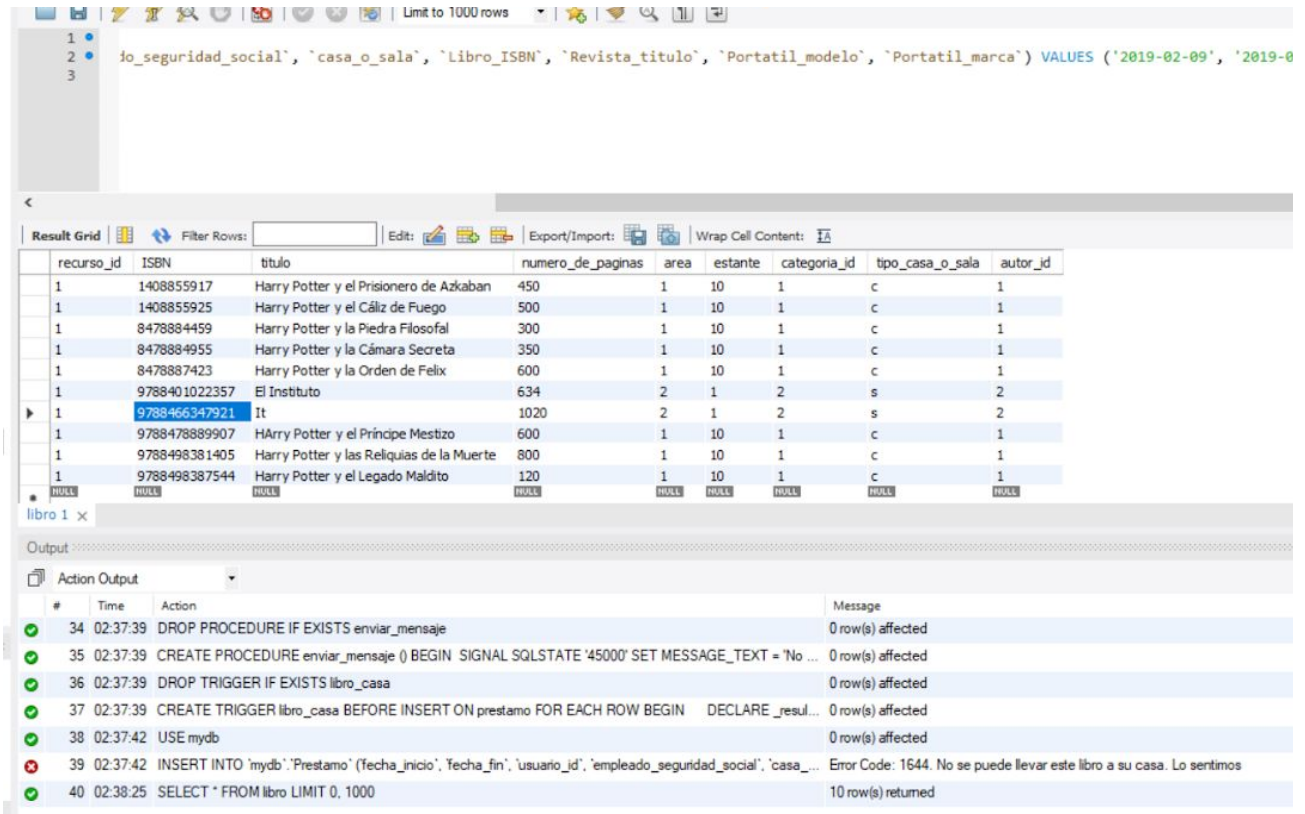
```



#	Time	Action	Message	Duration / F
33	02:36:45	INSERT INTO 'mydb'.'Prestamo' ('Fecha_inicio', 'Fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_...	1 row(s) affected	0.062 sec
34	02:37:39	DROP PROCEDURE IF EXISTS enviar_mensaje	0 row(s) affected	0.125 sec
35	02:37:39	CREATE PROCEDURE enviar_mensaje () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No ...	0 row(s) affected	0.047 sec
36	02:37:39	DROP TRIGGER IF EXISTS libro_casa	0 row(s) affected	0.063 sec
37	02:37:39	CREATE TRIGGER libro_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _resul...	0 row(s) affected	0.109 sec
38	02:37:42	USE mydb	0 row(s) affected	0.000 sec
39	02:37:42	INSERT INTO 'mydb'.'Prestamo' ('Fecha_inicio', 'Fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_...	Error Code: 1644. No se puede llevar este libro a su casa. Lo sentimos	0.000 sec

 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

Vemos que efectivamente este libro es solo para la sala y no lo deja reservar:



SQL Query:

```

INSERT INTO `libro` (`recurso_id`, `ISBN`, `titulo`, `numero_de_paginas`, `area`, `estante`, `categoria_id`, `tipo_casa_o_sala`, `autor_id`) VALUES ('2019-02-09', '2019-0

```

recurso_id	ISBN	titulo	numero_de_paginas	area	estante	categoria_id	tipo_casa_o_sala	autor_id
1	1408855917	Harry Potter y el Prisionero de Azkaban	450	1	10	1	c	1
1	1408855925	Harry Potter y el Cáliz de Fuego	500	1	10	1	c	1
1	8478884459	Harry Potter y la Piedra Filosofal	300	1	10	1	c	1
1	8478884955	Harry Potter y la Cámara Secreta	350	1	10	1	c	1
1	8478887423	Harry Potter y la Orden de Felix	600	1	10	1	c	1
1	9788401022357	El Instituto	634	2	1	2	s	2
1	9788466347921	It	1020	2	1	2	s	2
1	9788478889907	Harry Potter y el Príncipe Mestizo	600	1	10	1	c	1
1	9788498381405	Harry Potter y las Reliquias de la Muerte	800	1	10	1	c	1
1	9788498387544	Harry Potter y el Legado Maldito	120	1	10	1	c	1
1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output:

```

# Time Action Message
34 02:37:39 DROP PROCEDURE IF EXISTS enviar_mensaje 0 row(s) affected
35 02:37:39 CREATE PROCEDURE enviar_mensaje () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No ... 0 row(s) affected
36 02:37:39 DROP TRIGGER IF EXISTS libro_casa 0 row(s) affected
37 02:37:39 CREATE TRIGGER libro_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _resul... 0 row(s) affected
38 02:37:42 USE mydb 0 row(s) affected
39 02:37:42 INSERT INTO `mydb`.`Prestamo` (`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa_... Error Code: 1644. No se puede llevar este libro a su casa. Lo sentimos
40 02:38:25 SELECT * FROM libro LIMIT 0, 1000 10 row(s) returned

```

5.- Las revistas se prestarán solo si hay fotocopias realizadas de la misma:

USE mydb:

```
delimiter //
```

```
DROP PROCEDURE IF EXISTS enviar_mensaje_revista //
```


```
CREATE PROCEDURE enviar_mensaje_revista ()
```

```
BEGIN
```

```

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede llevar esta revista
a su casa. Debe realizar las fotocopias y llevarselas.. Lo sentimos';

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

END//

delimiter ;

DELIMITER \$\$

DROP TRIGGER IF EXISTS revista_casa \$\$

CREATE TRIGGER revista_casa

BEFORE INSERT ON prestamo FOR EACH ROW

BEGIN

 DECLARE _result BOOL DEFAULT FALSE;

 if ((SELECT fotocopias_realizadas FROM revista WHERE NEW.Revista_titulo
= revista.titulo) = FALSE) THEN

 SET _RESULT=TRUE;

 END IF;


 if (_result) THEN

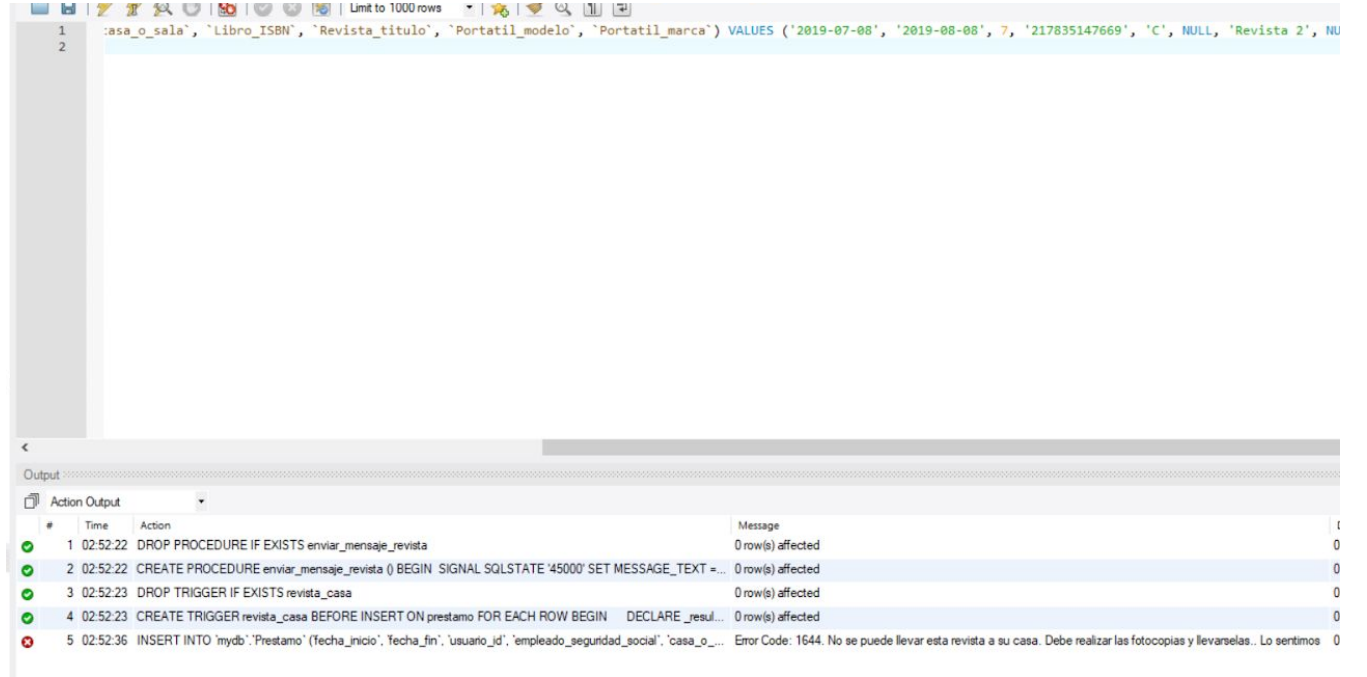
 CALL enviar_mensaje_revista();

 END IF;

END \$\$

DELIMITER ;

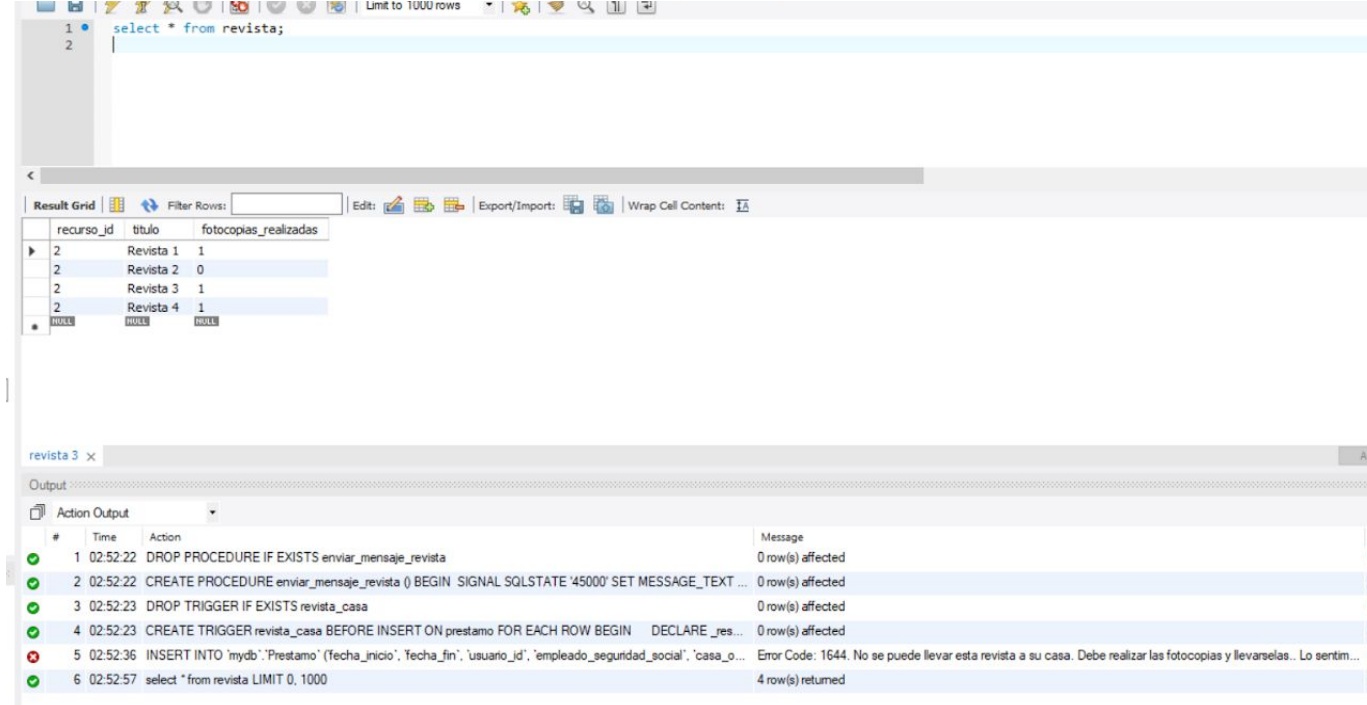
	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20



Output

#	Time	Action	Message	
1	02:52:22	DROP PROCEDURE IF EXISTS enviar_mensaje_revista	0 row(s) affected	0
2	02:52:22	CREATE PROCEDURE enviar_mensaje_revista () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =...	0 row(s) affected	0
3	02:52:23	DROP TRIGGER IF EXISTS revista_casa	0 row(s) affected	0
4	02:52:23	CREATE TRIGGER revista_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _resul...	0 row(s) affected	0
5	02:52:36	INSERT INTO 'mydb'. 'Prestamo' ('fecha_inicio', 'fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_o_...	Error Code: 1644. No se puede llevar esta revista a su casa. Debe realizar las fotocopias y llevarsela... Lo sentimos	0

Esto ocurre porque la revista 2 no tiene las fotocopias realizadas.




Result Grid

recurso_id	titulo	fotocopias_realizadas
2	Revista 1	1
2	Revista 2	0
2	Revista 3	1
2	Revista 4	1
NULL	NULL	NULL

Output

#	Time	Action	Message
1	02:52:22	DROP PROCEDURE IF EXISTS enviar_mensaje_revista	0 row(s) affected
2	02:52:22	CREATE PROCEDURE enviar_mensaje_revista () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =...	0 row(s) affected
3	02:52:23	DROP TRIGGER IF EXISTS revista_casa	0 row(s) affected
4	02:52:23	CREATE TRIGGER revista_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _res...	0 row(s) affected
5	02:52:36	INSERT INTO 'mydb'. 'Prestamo' ('fecha_inicio', 'fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_o_...	Error Code: 1644. No se puede llevar esta revista a su casa. Debe realizar las fotocopias y llevarsela... Lo sentimos
6	02:52:57	select * from revista LIMIT 0, 1000	4 row(s) returned

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU010104373 9
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

6.- Los portátiles se prestarán directamente pero se deberán almacenar distintos datos al ser un objeto de mayor valor y tener más control sobre él.

USE mydb:

```

delimiter //
DROP PROCEDURE IF EXISTS enviar_mensaje_portatil //
CREATE PROCEDURE enviar_mensaje_portatil ()
BEGIN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Lo sentimos, no podemos darle
el portatil, tiene que enviarnos su DNI antes.';

END//

delimiter ;

DELIMITER $$
DROP TRIGGER IF EXISTS portatil_casa $$
CREATE TRIGGER portatil_casa
BEFORE INSERT ON prestamo FOR EACH ROW
BEGIN

    DECLARE _result BOOL DEFAULT TRUE;

    IF ((SELECT DNI FROM usuario WHERE NEW.usuario_id = usuario.id)) THEN
        SET _RESULT=false;


    END IF;

    if (_result) THEN
        CALL enviar_mensaje_portatil();

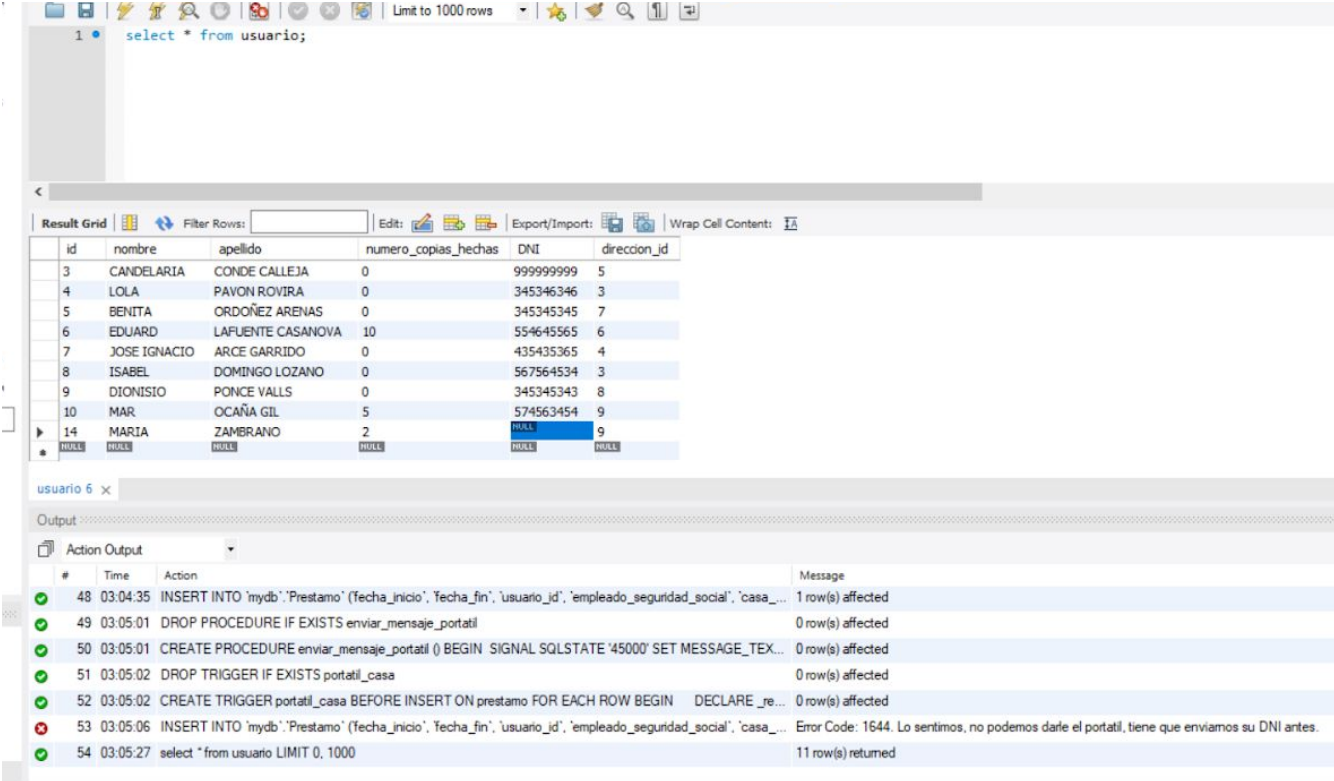
    END IF;

END $$
DELIMITER ;

```


 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
	PROYECTO: Biblioteca		Generación de código
	Autor:Alu0101043739		ALU0101043739
Versión: 2.0		Tiempo invertido:20 h	Fecha : 25/01/20

Vemos que el usuario 14 no tiene su dni establecido :



The screenshot shows a database management interface. At the top, a query is entered: `select * from usuario;`. Below the query, a table displays the results of the query. The table has columns: `id`, `nombre`, `apellido`, `numero_copias_hechas`, `DNI`, and `direccion_id`. The data rows show users 3 through 14. User 14, MARIA ZAMBRANO, has a `DNI` value of `NULL`. Below the table, an 'Action Output' log shows the execution of various SQL commands. The log includes an error message for user 14: 'Error Code: 1644. Lo sentimos, no podemos darle el portatil, tiene que enviarnos su DNI antes.'

id	nombre	apellido	numero_copias_hechas	DNI	direccion_id
3	CANDELARIA	CONDE CALLEJA	0	999999999	5
4	LOLA	PAVON ROVIRA	0	345346346	3
5	BENITA	ORDÓÑEZ ARENAS	0	345345345	7
6	EDUARD	LAFUENTE CASANOVA	10	554645565	6
7	JOSE IGNACIO	ARCE GARRIDO	0	435435365	4
8	ISABEL	DOMINGO LOZANO	0	567564534	3
9	DIONISIO	PONCE VALLS	0	345345343	8
10	MAR	OCAÑA GIL	5	574563454	9
14	MARIA	ZAMBRANO	2	NULL	9
*	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
48	03:04:35	INSERT INTO 'mydb'. 'Prestamo' ('fecha_inicio', 'fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_...	1 row(s) affected
49	03:05:01	DROP PROCEDURE IF EXISTS enviar_mensaje_portatil	0 row(s) affected
50	03:05:01	CREATE PROCEDURE enviar_mensaje_portatil () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEX...	0 row(s) affected
51	03:05:02	DROP TRIGGER IF EXISTS portatil_casa	0 row(s) affected
52	03:05:02	CREATE TRIGGER portatil_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _re...	0 row(s) affected
53	03:05:06	INSERT INTO 'mydb'. 'Prestamo' ('fecha_inicio', 'fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_...	Error Code: 1644. Lo sentimos, no podemos darle el portatil, tiene que enviarnos su DNI antes.
54	03:05:27	select * from usuario LIMIT 0, 1000	11 row(s) returned