



## Tabulación

Jugador(nick,mail,NomYApe,fechaNac,edad)

Partida(idpartida,nickjugador1,nickJugador2,resultado,estado,fecha,horaInicio,horaFinfila,columna)

Celda(x,y)

Tiene (cx,cy,idpart)

nickjugador1, nickjugador2: CF que referencia a Jugador

cx,cy: CF que referencia a Celda

idpart: CF que referencia a Partida

# CODIGO

```
DROP TABLE IF EXISTS tiene;

DROP TABLE IF EXISTS celda;

DROP TABLE IF EXISTS auditoria_Eliminados;

DROP TABLE IF EXISTS partida;

DROP TABLE IF EXISTS jugador;

DROP TRIGGER IF EXISTS chequeo_solapamiento;

DROP TRIGGER IF EXISTS eliminar_jugador;

DROP TRIGGER IF EXISTS cheq_tamceld;
```

```
/*Tabla de jugadores*/
```

```
CREATE TABLE jugador(

    id int NOT NULL AUTO_INCREMENT,

    nick VARCHAR(20),

    email VARCHAR(20),

    nombreAPellido VARCHAR(30),

    fechaNac DATE,

    edad int, /* calculado */

    UNIQUE (nick,email),

    CONSTRAINT jugador_pk PRIMARY KEY (id)

)ENGINE=InnoDB;
```

```
CREATE TABLE partida(

    idpartida INT NOT NULL PRIMARY KEY AUTO_INCREMENT,

    resultado ENUM("Ganador","Empate","Perdedor"),

    estado bool,

    fecha DATE,

    fila INT,

    columna INT,

    horaInicio VARCHAR (10),

    horaFin VARCHAR (10),

    id1 int,
```

```

        id2 int,

        CONSTRAINT fk_id1 FOREIGN KEY (id1) REFERENCES jugador(id) ON DELETE
        CASCADE, /**/

        CONSTRAINT fk_id2 FOREIGN KEY (id2) REFERENCES jugador(id) ON DELETE
        CASCADE

    )ENGINE=InnoDB;

CREATE TABLE celda(

    filx INT,

    coly INT,

    PRIMARY KEY (filx,coly)

)ENGINE=InnoDB;

CREATE TABLE tiene(

    num_mov INT NOT NULL PRIMARY KEY , /*numero de movimientos */

    idpartida INT,

    idx INT,

    idy INT,

    CONSTRAINT fk_id_partida FOREIGN KEY (idpartida) REFERENCES partida
    (idpartida) ON DELETE CASCADE,

    CONSTRAINT fk_idx FOREIGN KEY (idx, idy) REFERENCES celda (filx, coly)
    ON DELETE CASCADE

)ENGINE=InnoDB;

/*Tabla de jugadores eliminados*/
CREATE TABLE auditoria_Eliminados(

    id int PRIMARY KEY,

    nickElim VARCHAR(20),

    fechaElim datetime,

    eliminadoPor VARCHAR(10) /*Quien elimino el jugador*/

)ENGINE=InnoDB;

/*TRIGGER que genera la tabla de Auditoria de jugadores Eliminados*/
delimiter $$

```

```

CREATE TRIGGER eliminar_jugador BEFORE DELETE on jugador

FOR EACH ROW

BEGIN

    INSERT INTO auditoria_Eliminados (id,nickElim,fechaElim,eliminadoPor)
VALUES (Old.id,Old.nick,SYSDATE(),CURRENT_USER());

END$$

delimiter ;

/*TRIGGER que controla que no se solapen las partidas*/

delimiter $$

CREATE TRIGGER chequeo_solapamiento BEFORE INSERT ON partida

FOR EACH ROW

BEGIN

    IF EXISTS ( SELECT fecha,horaInicio,horaFin,id1,id2 FROM partida

                WHERE fecha=NEW.fecha AND ( (new.id1=id1 or new.id1=id2 )

                OR (new.id2=id1 or new.id2=id2) ) /*Para igual fecha, ver las
horas disponibles que tiene para jugar*/

                AND (horaInicio<=NEW.horaInicio AND NEW.horaInicio<=horaFin
/*primer caso que la hora nueva este dentro de las hs q ya tiene*/

                AND NEW.horaFin>=horaInicio AND NEW.horaFin<=horaFin)

                OR (NEW.horaInicio<horaInicio AND NEW.horaFin<horaFin AND
NEW.horaFin>=horaInicio ) /*2:Q la hora inicio menor pero la mayor dentro */

                OR (NEW.horaFin>horaFin AND NEW.horaInicio>horaInicio AND
NEW.horaInicio<horaFin )/*3:q la hora inicio dentro, la mayor fuera*/

                OR (NEW.horaInicio<horaInicio AND NEW.horaFin>horaFin) ) /*que
ambas inicio y fin este fuera newhoraInicio[horaIniciojugadahoraFin]
newHoraFin */

    THEN

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "UD NO PUEDE JUGAR :
DEBE FINALIZAR SU PARTIDA ANTERIOR. " ;

    END IF;

END$$

delimiter ;

/*TRIGGER que controla que el movimiento insertado este dentro del tablero*/

delimiter $$

```

```

CREATE TRIGGER cheq_tamceld BEFORE INSERT ON tiene
FOR EACH ROW
BEGIN
    DECLARE f INT;

    DECLARE c INT;

    SET f = (SELECT SUM(fila) FROM partida WHERE idpartida=NEW.idpartida);

    SET c = (SELECT SUM(columna) FROM partida WHERE
idpartida=NEW.idpartida);

    IF (((NEW.idx>=0) AND (NEW.idx<=f)) AND (((NEW.idy>=0) AND (NEW.idy<=c))))
THEN
        INSERT INTO celda (filx,coly)VALUES(NEW.idx,NEW.idy);

    ELSE

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = " POSICION INCORRECTA
" ;

    END IF;

    END$$

delimiter ;

```

```

/*Inserta en la tabla jugadores*/

```

```

INSERT INTO jugador (id,nick,email,nombreAPellido,fechaNac,edad) VALUES

(1,'mauri','mauri@gmail.com','Mauricio Delle','1986-05-05',28),

(2,'Fran','fran@gmail.com','Fran Roig','1993-03-03',22),

(3,'Matias','matu_dmr@hotmail.com','Matias Rondeau','1986-09-01',28),

(4,'jugadorPrueba1','prueba1@gmail.com','Jugador Prueba1','1993-03-06',21),

(5,'jugadorPrueba2','prueba2@gmail.com','Jugador Prueba2','1970-03-04',45);

```

```

/*Inserta en la tabla partida*/

```

```

INSERT INTO
partida(idpartida,resultado,estado,fecha,fila,columna,horaInicio,horaFin,id1,i
d2) VALUES

(3,'Empate',true,'2015-02-02',8,8,'15:00','16:00',1,2),

(4,'Pierde',true,'2015-02-02',8,8,'15:00','16:00',1,2),

(5,'Ganador',true,'2015-02-03',8,7,'14:00','14:15',3,4);

```

```
/*Inserta en la tabla tiene*/  
  
INSERT INTO tiene (num_mov,idpartida,idx,idy) VALUES  
  
(1,4,3,5),  
  
(1,5,4,4),  
  
(1,3,1,2);
```

## INTEGRANTES

Rondeau, Matias.

Roig, Francisco.

Delle Vedove, Mauricio.