

EJERCICIO 4: CHEESEBRIDGE

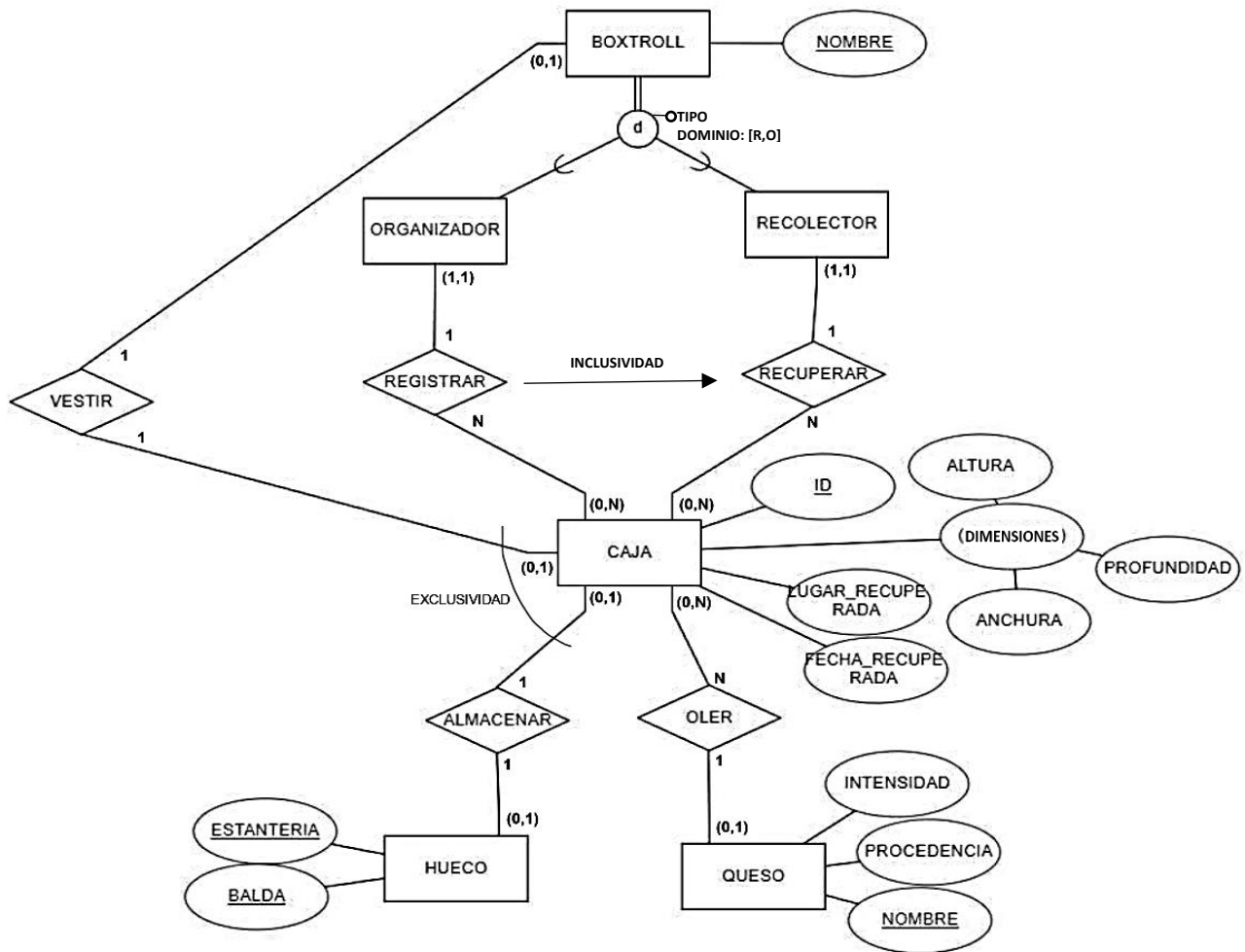
Pablo Fernández del Amo. 2022.

Cheesebridge (Puentequeso) es una elegante ciudad victoriana obsesionada por el dinero, la clase y el más delicioso de los quesos apestosos. Debajo de sus calles adoquinadas habitan los Boxtrolls, unos monstruos estrafalarios y adorables que visten cajas de cartón reciclado igual que las tortugas llevan su caparazón. Cada noche los boxtrolls salen de las alcantarillas de Cheesebridge para recoger los envases de cartón que sus habitantes desechan. Las cajas recuperadas se clasifican y almacenan cuidadosamente, pero la enorme cantidad acumulada en los últimos tiempos comienza a dificultar esta tarea. Atún es un boxtroll con algunas nociones sobre diseño de bases de datos relacionales, que desea almacenar la información siguiente:

1. Los boxtrolls forman una comunidad bien avenida. Todos se conocen por su nombre propio y se cuidan entre sí. Una distribución justa de las tareas asegura el correcto funcionamiento de la colonia, ya que cada monstruo realiza los trabajos para los que resulta más competente. En este aspecto, existen dos tipos de boxtrolls: los recolectores (aquéllos que salen cada noche para conseguir material de reciclaje) y los organizadores (aquéllos que se encargan de la clasificación y el almacenaje de lo conseguido).
2. Los recolectores recuperan, entre otros objetos, cajas de cartón de las que los organizadores registran sus dimensiones (altura, anchura y profundidad), además del lugar y fecha del hallazgo. Una vez examinadas, las cajas se guardan en el almacén o visten a los boxtrolls. De las cajas almacenadas se indican los números de estantería y balda en que se guardan. Por ejemplo, Tacón encontró en el colegio de Cheesebridge dos cajas de queso camembert de dimensiones 40×40×40 el día 10/11/2014. Dichas cajas fueron procesadas por el organizador Specs que guardó una de ellas en el almacén (estantería 4 - balda 7) y registró la otra como vestimenta de Voltio.
3. Muchos envases reciclados de Cheesebridge han contenido queso. Algunos tienen un aroma muy peculiar y por ello los organizadores huelen cada caja y toman nota del tipo de queso más oloroso que ha almacenado (procedencia e intensidad). Por ejemplo, entre los quesos más apestosos se encuentran el vieux-Boulogne (Francia, intensidad 10) y el pont-l'Eveque (Francia, intensidad 9), mientras que el cheddar (Reino Unido, intensidad 3) y el parmesano (Italia, intensidad 2) son los quesos menos aromáticos.

1. MODELO ENTIDAD – RELACIÓN

1.1. MODELO



1.2. JUSTIFICACIONES DE DISEÑO

- En la entidad CAJA encontramos todas las cajas que han sido recuperadas por un recolector (RECOLECTOR) y organizadas por un organizador (ORGANIZADOR). La restricción del modelo extendido de INCLUSIVIDAD expone la restricción de que una caja se ha registrado si y solo si ha sido recuperada previamente. Aunque con las cardinalidades ya tenemos que cada caja existente ha sido recuperada y registrada, completamos la información de esta forma.
- La entidad BOXTROLL tiene dos subtipos (relación de jerarquía total disjunta).
- Si una caja ha sido registrada por un organizador, y por ende se han registrado sus dimensiones, fecha y lugar donde se recuperó. También, a través de las relaciones con BOXTROLL (VESTIR), HUECO (ALMACENAR) y QUESO (OLER), indicamos si han sido usadas como vestimenta, almacenadas o contenían qué tipo de queso; siendo las dos primeras incompatibles al mismo tiempo (de ahí la exclusividad marcada).
- Los BOXTROLL pueden estar sin vestimenta asociada (diseñado así para que en el diseño físico podamos introducir tuplas en la tabla BOXTROLL y en la tabla CAJA) y solo puede tener una caja como vestimenta. De ahí la cardinalidad (0,1). Si

obligáramos a cada boxtroll a estar con una caja como vestimenta, al introducir un nuevo boxtroll en la entidad, deberíamos haber asignado una caja ya a ese boxtroll, pero sin su nombre como referencia no lo podemos asignar.

- En la entidad QUESO guardamos el catálogo de quesos. Esta entidad está relacionada con las cajas que olían a queso a través de la relación OLER. No todas las cajas contienen y huelen a queso. Cada caja contiene a lo sumo un queso. De ahí las cardinalidades.
- La clave primaria de la entidad HUECO es compuesta por los atributos BALDA y ESTANTERIA, formando su combinación la clave primaria del hueco. Una caja solo puede ser almacenada en un único hueco. Un hueco solo tiene espacio para una caja.

2. MODELO RELACIONAL

Las justificaciones de diseño las haremos a medida que creamos el modelo.

- **RELACIONES**

BOXTROLL(NOMBRE, TIPO)

CAJA(ID, FECHA_RECUPERADA, LUGAR_RECUPERADA, PROFUNDIDAD, ALTURA, ANCHURA, NOMBRE_RECOLECTOR, NOMBRE_ORGANIZADOR, NOMBRE_VISTE*)

Añadiendo los campos NOMBRE_RECOLECTOR y NOMBRE_ORGANIZADOR no nulos cumplimos con las cardinalidades que informan de que cada caja ha sido recuperada y registrada por dos boxtrolls (cada uno del tipo correspondiente), y con la restricción de inclusividad marcada en el e/r. Con NOMBRE_VISTE referenciamos el boxtroll que utiliza esa caja para vestir. Puede ser nulo ya que hay cajas que se almacenan.

QUESO(NOMBRE, PROCEDENCIA, INTENSIDAD)

HUECO(BALDA, ESTANTERIA)

CAJA_CON_QUESO(ID_CAJA, NOMBRE_QUESO)

CAJA_ALMACENADA(ID_CAJA, BALDA_HUECO, ESTANTERIA_HUECO)

También, está normalizado, ya que la relación está en FNBC (aunque no visto en clase, está en 5 FN), de ahí la aparición de ciertas tablas como QUESO o HUECO.

- **CLAVES AJENAS**

CAJA(NOMBRE_RECOLECTOR) -> BOXTROLL(NOMBRE)

- BORRADO: RESTRICT | MODIFICACION: CASCADE

- *No se puede borrar un boxtroll que haya recuperado una caja, ya que el campo NOMBRE_RECOLECTOR está diseñado para no aceptar valores nulos. Por tanto, almacenamos en la relación BOXTROLL también aquellos que han fallecido. Si no, cambiamos el campo a NOMBRE_RECOLECTOR*, cambiando también las cardinalidades en el modelo E/R*

CAJA(NOMBRE_ORGANIZADOR) -> BOXTROLL(NOMBRE)

- BORRADO: RESTRICT | MODIFICACION: CASCADE

- *No se puede borrar un boxtroll que haya registrado una caja, ya que el campo NOMBRE_ORGANIZADOR está diseñado para no aceptar valores nulos. Por tanto, almacenamos en la relación BOXTROLL también aquellos que han fallecido. Si no, cambiamos el campo a NOMBRE_ORGANIZADOR*, cambiando también las cardinalidades en el modelo E/R*

CAJA(NOMBRE_VISTE) -> BOXTROLL(NOMBRE)

- BORRADO: SET NULL | MODIFICACION: CASCADE

CAJA_CON_QUESO(ID_CAJA) -> CAJA(ID)

- BORRADO: CASCADE | MODIFICACION: CASCADE

CAJA_CON_QUESO(NOMBRE_QUESO) -> QUESO(NOMBRE)

- BORRADO: RESTRICT | MODIFICACION: CASCADE

RESTRICT ya que no se puede borrar un queso si no se han borrado previamente todas las cajas que olían a ese queso.

CAJA_ALMACENADA(ID_CAJA) -> CAJA(ID)

- BORRADO: CASCADE | MODIFICACION: CASCADE

CAJA_ALMACENADA(BALDA_HUECO, ESTANTERIA_HUECO) -> HUECO(BALDA, ESTANTERIA)

- BORRADO: RESTRICT | MODIFICACION: CASCADE

RESTRICT ya que solo puedo eliminar un hueco si este está vacío, así que antes de borrarlo la caja que contiene tiene que ser recolocada o entregada a un boxtroll como vestimenta.

• REQUISITOS NO INCLUIDOS EN EL ESQUEMA

- El dominio del campo TIPO en BOXTROLL solo puede ser O (Organizador) o R (recolector), ya que todos los boxtrolls se dedican a solo una de estas tareas. No se contempla que un boxtroll pueda cambiar de oficio – ya que el enunciado no especifica que se pueda. En este caso, al hacer el modelo físico deberemos eliminar el check que obliga a que los campos de CAJA NOMBRE_RECOLECTOR y NOMBRE_ORGANIZADOR referencien a boxtroll recolectores y organizadores, respectivamente.
- La clave foránea (NOMBRE_ORGANIZADOR) en CAJA referencia la clave primaria NOMBRE de la relación BOXTROLL, pero solo son válidas aquellos nombres pertenecientes a tuplas con TIPO O, exclusivamente. (solo registran los organizadores)
- La clave foránea (NOMBRE_RECOLECTOR) en CAJA referencia la clave primaria NOMBRE de la relación BOXTROLL, pero solo son válidas aquellos nombres pertenecientes a tuplas con TIPO R, exclusivamente. (solo registran los recolectores)
- No aparecen entradas en la relación CAJA_ALMACENADA cuyo ID_CAJA referencia a un registro de la relación CAJA con campo NOMBRE_VISTE no nulo, y viceversa. Ya que ambas relaciones, como se ve en el E/R en la restricción de exclusividad. Toda entrada en la entidad CAJA tiene el campo NOMBRE_VISTE no nulo o aparece en la relación CAJA_ALMACENADA. *Se ha contemplado el caso de que haya cajas que todavía no se ha decidido si almacenarlas o entregarlas como vestimenta. En el caso de que no contemplásemos esta posibilidad, bastaría con añadir en la relación caja los campos balda y hueco, obligando a que sean nulos solo si nombre_viste no lo es y viceversa.*
- El total de cajas registradas debe ser menor al total de boxtrolls más el total de huecos, ya que el enunciado expresa claramente que las dos opciones que tiene una caja registrada es servir como vestimenta para un boxtroll o ser almacenada en un hueco. No puede haber una caja de la relación CAJA que no aparezca como

almacenada en la relación CAJA_ALMACENADA o como vestimenta en la relación BOXTROLL.

3. DISEÑO FÍSICO E IMPLEMENTACIÓN

A continuación, se puede estudiar mi código de SQL para la creación de esta base de datos y pantallazos de las consultas por orden:

```
/* CHEESEBRIDGE: DISEÑO FÍSICO E IMPLEMENTACIÓN. PABLO FERNÁNDEZ DEL AMO.
```

Los datos almacenados en la base de datos son acorde los datos expuestos en el enunciado, y acorde con los protagonistas y lugares de la película.

A continuación, expongo la creación de la base de datos relacional. Tanto la creación de tablas, identificación de claves, triggers, checks, inserciones... se han sentenciado en orden, para su lectura de forma clara: */

```
-- Creación y uso de la base de datos cheesebridge:
```

```
CREATE DATABASE IF NOT EXISTS cheesebridge;
```

```
USE cheesebridge;
```

```
-- Creación de tablas:
```

```
CREATE TABLE IF NOT EXISTS boxtroll (
```

```
    nombre VARCHAR(15) NOT NULL,
```

```
    tipo ENUM('O','R') NOT NULL
```

```
);
```

```
CREATE TABLE IF NOT EXISTS caja(
```

```
    id CHAR(6) NOT NULL,
```

```
    lugar_recuperada VARCHAR(30) NOT NULL,
```

```
    fecha_recuperada DATE NOT NULL,
```

```
    profundidad INT NOT NULL,
```

```
    altura INT NOT NULL,
```

```
    anchura INT NOT NULL,
```

```
    nombre_recolector VARCHAR(15) NOT NULL,
```

```
    nombre_organizador VARCHAR(15) NOT NULL,
```

```
    nombre_viste VARCHAR(15) NULL
```

```
);
```

```
CREATE TABLE IF NOT EXISTS queso(
```

```
    nombre VARCHAR(30) NOT NULL,
```

```
    procedencia VARCHAR(30) NOT NULL,
```

```
    intensidad INT NOT NULL
```

```
);
```

```
CREATE TABLE IF NOT EXISTS hueco(
```

```
    balda INT NOT NULL,
```

```
    estanteria INT NOT NULL
```

```
);
```

```
CREATE TABLE IF NOT EXISTS caja_con_queso(
```

```

        id_caja CHAR(6) NOT NULL,
        nombre_queso VARCHAR(30) NOT NULL
    );
CREATE TABLE IF NOT EXISTS caja_almacenada(
        id_caja CHAR(6) NOT NULL,
        balda_hueco INT NOT NULL,
        estanteria_hueco INT NOT NULL
    );

-- Identificación de las claves primarias de las tablas creadas:
ALTER TABLE boxtroll
    ADD CONSTRAINT boxtrollpk PRIMARY KEY (nombre);
ALTER TABLE caja
    ADD CONSTRAINT cajapk PRIMARY KEY (id);
ALTER TABLE queso
    ADD CONSTRAINT quesopk PRIMARY KEY (nombre);
ALTER TABLE hueco
    ADD CONSTRAINT huecopk PRIMARY KEY (balda,estanteria);
ALTER TABLE caja_con_queso
    ADD CONSTRAINT caja_con_quesopk PRIMARY KEY (id_caja);
ALTER TABLE caja_almacenada
    ADD CONSTRAINT caja_almacenadapk PRIMARY KEY (id_caja);

-- Identificación del resto de claves candidatas (únicas):
ALTER TABLE caja_almacenada
    ADD CONSTRAINT u_caja_almacenada UNIQUE (balda_hueco,estanteria_hueco);

-- Identificación de las claves foráneas de las tablas creadas:
ALTER TABLE caja
    ADD CONSTRAINT cajafk1 FOREIGN KEY (nombre_recolector)
    REFERENCES boxtroll(nombre)
    ON DELETE RESTRICT ON UPDATE CASCADE;
ALTER TABLE caja
    ADD CONSTRAINT cajafk2 FOREIGN KEY (nombre_organizador)
    REFERENCES boxtroll(nombre)
    ON DELETE RESTRICT ON UPDATE CASCADE;
ALTER TABLE caja
    ADD CONSTRAINT cajafk3 FOREIGN KEY (nombre_viste)
    REFERENCES boxtroll(nombre)
    ON DELETE SET NULL ON UPDATE CASCADE;
ALTER TABLE caja_con_queso
    ADD CONSTRAINT caja_con_quesofk1 FOREIGN KEY (id_caja)
    REFERENCES caja(id)
    ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE caja_con_queso
    ADD CONSTRAINT caja_con_quesofk2 FOREIGN KEY (nombre_queso)
    REFERENCES queso(nombre)
    ON DELETE RESTRICT ON UPDATE CASCADE;

```

```

ALTER TABLE caja_almacenada
    ADD CONSTRAINT caja_almacenadafk1 FOREIGN KEY (id_caja)
    REFERENCES caja(id)
    ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE caja_almacenada
    ADD CONSTRAINT caja_almacenadafk2 FOREIGN KEY
(balda_hueco,estanteria_hueco)
    REFERENCES hueco(balda,estanteria)
    ON DELETE RESTRICT ON UPDATE CASCADE;

```

/* Creación de CHECKs:

La primera idea fue crear estos dos checks, pero como MySQL no acepta subqueries en los checks, los transformamos en triggers más adelante:

```

ALTER TABLE CAJA
ADD CHECK (
    (SELECT COUNT(*) FROM CAJA) < (SELECT COUNT(*) FROM BOXTROLL) + (SELECT COUNT(*)
FROM CAJA_ALMACENADA)
);

```

```

ALTER TABLE CAJA
ADD CONSTRAINT check_caja
CHECK (EXISTS(SELECT * FROM BOXTROLL WHERE NOMBRE = nombre_organizador AND
TIPO = 'O') AND
    EXISTS(SELECT * FROM BOXTROLL WHERE NOMBRE = nombre_recolector AND TIPO =
'R'));
*/

```

-- La intensidad máxima y mínima es 10 y 0 respectivamente. Exigimos con el check que así sea:

```

ALTER TABLE queso
ADD CONSTRAINT chk_queso CHECK (intensidad>=0 and intensidad<11);

```

-- No podemos aceptar posiciones con valores negativos.

```

ALTER TABLE hueco
ADD CONSTRAINT chk_hueco CHECK (balda>=0 and estanteria>=0);

```

-- No podemos aceptar dimensiones con valores negativos.

```

ALTER TABLE caja
ADD CONSTRAINT chk_caja CHECK (profundidad>=0 and anchura>=0 and altura>=0);

```

-- Creación de TRIGGERS útiles:

```

DELIMITER $$

```

/* Toda caja registrada debe estar almacenada o ser utilizada como vestimenta. Si todos los huecos están ocupados por cajas

y todos los boxtrolls vestidos, no podremos recuperar ni registrar ninguna caja más: */

```

CREATE TRIGGER limite_al_insertar_en_caja
BEFORE INSERT ON caja
FOR EACH ROW
BEGIN

```

```

DECLARE total INTEGER;
SET total = (SELECT COUNT(*) FROM boxtroll)+(SELECT COUNT(*) FROM caja_almacenada);
IF total <= (SELECT COUNT(*) FROM caja) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'No se pueden insertar nuevos registros en la tabla CAJA porque al
no haber suficientes registros en la tabla HUECO ni en la BOXTROLL';
END IF;
END$$

```

```

/* Antes de insertar nuevos registros en CAJA, tenemos que asegurarnos que el boxtroll
introducido como recolector es de tipo R
y el organizador de tipo O: */
CREATE TRIGGER tipo_o_r
BEFORE INSERT ON caja
FOR EACH ROW
BEGIN
    IF (SELECT tipo FROM boxtroll WHERE nombre = NEW.nombre_organizador) <> 'O' OR
    (SELECT tipo FROM BOXTROLL WHERE nombre = NEW.nombre_recolector) <> 'R' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El nombre_organizador debe referenciar a un boxtroll de tipo O y
el nombre_recolector debe referenciar a un boxtroll de tipo R';
    END IF;
END$$

```

```

/* Si se modifica un registro de la tabla caja, cambiando su valor NULL del campo
NOMBRE_VISTE, implica que esa caja se utiliza como vestimenta,
debiendo borrarla de la tabla CAJA_ALMACENADA: */
CREATE TRIGGER borrar_tuplas_caja_almacenada
AFTER UPDATE ON caja
FOR EACH ROW
BEGIN
    DELETE FROM caja_almacenada WHERE id_caja = OLD.id AND OLD.nombre_viste IS NULL
AND NEW.nombre_viste IS NOT NULL;
END$$

```

/* Hay que comprobar que cuando se inserte un nuevo registro de caja en la tabla CAJA, no se asocie esta a un boxtroll que ya tenga caja asignada. */

```

CREATE TRIGGER verificar_nombre_viste
BEFORE INSERT ON caja
FOR EACH ROW
BEGIN
    IF (NEW.nombre_viste IS NOT NULL AND EXISTS (SELECT 1 FROM caja WHERE
nombre_viste = NEW.nombre_viste)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Un boxtroll no puede tener asignadas
dos cajas como vestimenta';
    END IF;

```


END\$\$

DELIMITER ;

/* Insercción de valores en cada tabla con datos importados de la película. Es decir, en su gran mayoría, estos datos son los que introduciría el responsable de la base de datos de los boxtrolls en la película: */

INSERT INTO boxtroll (nombre, tipo) VALUES

("Eggs", "O"),
("Specs", "O"),
("Oilcan", "O"),
("Archie", "O"),
("Shoe", "O"),
("Fish", "O"),
("Mr. Gristle", "O"),
("Mr. Trout", "O"),
("Mr. Pickles", "O"),
("Mr. Snatcher", "O"),
("Mr. Gristle Jr.", "O"),
("Tin", "O"),
("Rubber", "O"),
("Tacón", "R"),
("Voltio", "R"),
("Wires", "R"),
("Shoe Jr.", "R"),
("Tug", "R"),
("Tops", "R"),
("Boots", "R"),
("Nacho", "R");

INSERT INTO caja(id, lugar_recuperada, fecha_recuperada, profundidad, altura, anchura, nombre_recolector, nombre_organizador, nombre_viste) VALUES

('CJ0001', 'Colegio de Cheesebridge', '2014-11-10', 40, 40, 40, 'Tacón', 'Specs', 'Voltio'),
('CJ0002', 'Barrio del puerto', '2014-11-11', 20, 50, 30, 'Tacón', 'Mr. Gristle', 'Specs'),
('CJ0003', 'Mercado de Cheesebridge', '2014-11-12', 30, 30, 30, 'Voltio', 'Fish', 'Tacón'),
('CJ0004', 'Calle principal', '2014-11-13', 10, 60, 20, 'Tacón', 'Mr. Trout', NULL),
('CJ0005', 'Parque del sur', '2014-11-14', 50, 20, 50, 'Nacho', 'Specs', 'Mr. Trout'),
('CJ0006', 'Barrio del centro', '2014-11-15', 30, 30, 30, 'Tacón', 'Mr. Gristle', 'Nacho'),
('CJ0007', 'Calle del oeste', '2014-11-16', 40, 40, 40, 'Voltio', 'Specs', NULL),
('CJ0008', 'Parque del norte', '2014-11-17', 20, 50, 30, 'Nacho', 'Mr. Trout', 'Fish'),
('CJ0009', 'Calle del este', '2014-11-18', 10, 60, 20, 'Boots', 'Fish', NULL),
('CJ0010', 'Calle del sur', '2014-11-19', 50, 20, 10, 'Tops', 'Archie', 'Tin'),
('CJ0011', 'Colegio de Cheesebridge', '2014-11-10', 40, 40, 40, 'Tacón', 'Specs', NULL),
('CJ0012', 'Calle principal', '2014-11-13', 50, 50, 50, 'Tops', 'Archie', NULL),
('CJ0013', 'Barrio del centro', '2014-11-15', 40, 40, 40, 'Tops', 'Fish', 'Tug'),
('CJ0014', 'Barrio del puerto', '2014-11-11', 30, 30, 30, 'Voltio', 'Mr. Trout', NULL),

```

('CJ0015', 'Calle del este', '2014-11-18', 20, 50, 30, 'Boots', 'Fish', 'Shoe Jr.'),
('CJ0016', 'Mercado de Cheesebridge', '2014-11-12', 10, 60, 20, 'Tacón', 'Mr. Trout', 'Wires'),
('CJ0017', 'Parque del sur', '2014-11-14', 50, 20, 50, 'Nacho', 'Specs', NULL),
('CJ0018', 'Parque del norte', '2014-11-17', 30, 30, 30, 'Voltio', 'Mr. Trout', 'Rubber'),
('CJ0019', 'Calle del oeste', '2014-11-16', 40, 40, 40, 'Boots', 'Specs', NULL),
('CJ0020', 'Calle del sur', '2014-11-19', 20, 50, 30, 'Tops', 'Fish', NULL);

```

```

INSERT INTO queso (nombre, procedencia, intensidad) VALUES

```

```

('Vieux-Boulogne', 'Francia', 10),
('Pont-l'Eveque', 'Francia', 9),
('Stinking bishop', 'Reino Unido', 8),
('Taleggio', 'Italia', 7),
('Epoisses de bourgogne', 'Francia', 6),
('Camembert', 'Francia', 5),
('Gorgonzola', 'Italia', 4),
('Cheddar', 'Reino Unido', 3),
('Parmesano', 'Italia', 2),
('Feta', 'Grecia', 1),
('Roquefort', 'Francia', 10),
('Bleu brie', 'Francia', 9),
('Boursault', 'Francia', 8),
('Boursin', 'Francia', 7),
('Brie de meaux', 'Francia', 6),
('Chabichou du poitou', 'Francia', 5),
('Chevre', 'Francia', 4),
('Coulommiers', 'Francia', 3),
('Crottin de chavignol', 'Francia', 2),
('Emmental', 'Suiza', 1);

```

```

INSERT INTO caja_con_queso (id_caja, nombre_queso) VALUES

```

```

('CJ0001', 'Camembert'),
('CJ0002', 'Pont-l'Eveque'),
('CJ0003', 'Stinking bishop'),
('CJ0004', 'Taleggio'),
('CJ0005', 'Epoisses de bourgogne'),
('CJ0007', 'Gorgonzola'),
('CJ0008', 'Cheddar'),
('CJ0009', 'Parmesano'),
('CJ0010', 'Feta'),
('CJ0011', 'Camembert'),
('CJ0012', 'Bleu brie'),
('CJ0013', 'Boursault'),
('CJ0014', 'Brie de meaux'),
('CJ0017', 'Roquefort');

```

```

INSERT INTO hueco (balda, estanteria) VALUES

```

```

(1, 1),
(1, 2),

```

```

(1, 3),
(1, 4),
(2, 1),
(2, 2),
(2, 3),
(2, 4),
(3, 1),
(3, 2),
(3, 3),
(3, 4),
(4, 1),
(4, 2),
(4, 3),
(4, 4),
(5, 1),
(5, 2),
(5, 3),
(5, 4),
(6, 1),
(6, 2),
(6, 3),
(6, 4),
(7, 1),
(7, 2),
(7, 3),
(7, 4);
INSERT INTO caja_almacenada(id_caja, balda_hueco, estanteria_hueco) VALUES
('CJ0004',2,3),
('CJ0007',7,1),
('CJ0009',4,4),
('CJ0011',7,4),
('CJ0012',5,2),
('CJ0014',6,3),
('CJ0017',3,4),
('CJ0019',1,1),
('CJ0020',1,2);

```

-- Consultas sobre la base de datos cheesebridge:

-- 1. TOP 3 RECOLECTORES con mayor afinidad para, dentro de las cajas con queso recogidas, recuperar aquellas con mayor intensidad en general:

```

SELECT nombre_recolector as nombre, avg(intensidad) as media_intensidad FROM
caja INNER JOIN caja_con_queso
ON caja_con_queso.id_caja = caja.id
INNER JOIN queso ON caja_con_queso.nombre_queso=queso.nombre
GROUP BY nombre_recolector
having media_intensidad >5
ORDER BY media_intensidad desc

```

limit 3;

-- 2. TOP 3 ORGANIZADORES con mayor afinidad para, dentro de las cajas con queso organizadas, organizar aquellas con mayor intensidad en general:
SELECT nombre_organizador as nombre, avg(intensidad) as media_intensidad FROM
caja INNER JOIN caja_con_queso
ON caja_con_queso.id_caja = caja.id
INNER JOIN queso ON caja_con_queso.nombre_queso=queso.nombre
GROUP BY nombre_organizador
having media_intensidad >5
ORDER BY media_intensidad desc
limit 3;

-- 3. Nombre de los Organizadores que más cajas han organizado:
select nombre_organizador as nombre, count(*) as numero_cajas from caja
group by nombre_organizador
having count(*) = (
select max(numero) from (select count(*) as numero from caja
group by nombre_organizador) as t);

-- 4. Nombre de los Recolectores que más cajas han recuperado:
select nombre_recolector as nombre, count(*) as numero_cajas from caja
group by nombre_recolector
having count(*) = (
select max(numero) from (select count(*) as numero from caja
group by nombre_recolector) as t);

-- 5. Queso que más ha aparecido en las cajas:
select nombre_queso as nombre, count(*) as numero_apariciones from caja_con_queso
group by nombre_queso
having count(*) = (
select max(numero) from (select count(*) as numero from caja_con_queso
group by nombre_queso) as t);

-- 6. Lista de boxtrolls con su tipo que no han recogido ni organizado ninguna caja, pero aún así están vestidos.
select nombre from
boxtroll left join caja on nombre=nombre_organizador or nombre= nombre_recolector
where id is null and nombre in (select nombre_viste from caja) ;

-- 7. Lista de cajas que no han sido registradas, pero aún no han sido almacenadas o entregadas a un boxtroll:
SELECT c.ID as ID_caja
FROM caja c
LEFT JOIN caja_almacenada ca ON c.ID = ca.ID_caja
WHERE ca.ID_caja IS NULL and c.Nombre_viste is NULL;

/*

Se ha intentado crear el siguiente Trigger para no tener que consultar las cajas que faltan por almacenar o entregar, ya que en teoría comprueba al insertar en la tabla caja_almacenada si al insertar valores ya no queda ninguna caja registrada por asignar a un boxtroll o por almacenar, arrojando siempre el mensaje de error especificado.

Por ello, lo hemos comentado para mostrar que se ha tenido en cuenta este hecho.

```
DELIMITER $$
CREATE TRIGGER trg_caja_almacenada_aviso
AFTER INSERT ON caja_almacenada
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT * FROM caja WHERE nombre_viste IS NULL AND id NOT IN (SELECT id_caja
FROM caja_almacenada)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Faltan cajas que no son usadas para
vestir por almacenar';
    END IF;
END$$ */
```

-- FIN

PABLO FERNÁNDEZ DEL AMO

1)

nombre	numero_cajas
Specs	6

2)

nombre	media_intensidad
Tacón	6.5000
Nacho	6.3333
Tops	6.0000





3)

nombre	media_intensidad
Mr. Gristle	9.0000
Fish	6.0000
Specs	6.0000





4)

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	nombre	numero_cajas			
▶	Tacón	6			

5)

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	nombre	numero_apariciones			
▶	Camembert	2			

6)

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	nombre				
▶	Rubber				
	Shoe Jr.				
	Tin				
	Tug				
	Wires				

7)

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	ID_caja				