

Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales

Bases de Datos

1ER CUATRIMESTRE 2011

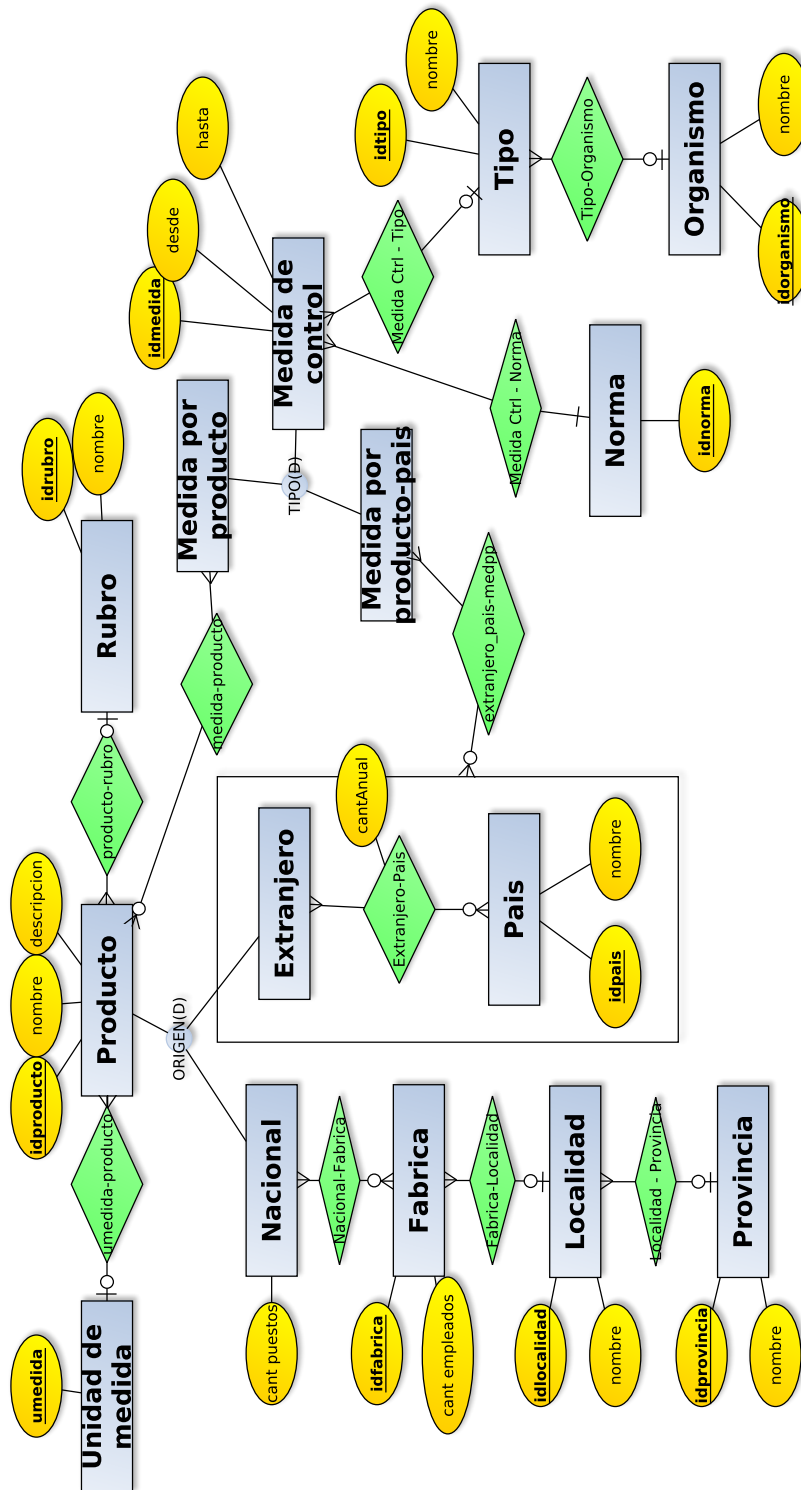
TP1

DER, MR, SQL

Mariano Bianchi	marianobianchi08@gmail.com
Pablo Brusco	pablo.brusco@gmail.com
Julian Dondero	juliandondero@gmail.com
Ezequiel Castellano	ezequiel.castellano@gmail.com
Kevin Allekotte	kevinalle@gmail.com

Mayo 2011

DER



Decisiones tomadas

1. Todos los productos tienen una Unidad de Medida asociada.
2. Pueden existir Unidades de Medida, sin Productos asociados.
3. Pueden existir Rubros, sin Productos asociados.
4. Pueden existir Fábricas sin Productos asociados.
5. Pueden existir Localidades sin Fábricas asociadas.
6. Pueden existir Provincias sin Localidades.
7. Pueden existir Países sin Productos asociados.
8. Pueden haber Productos, sin Medidas de Control asociadas.
9. No pueden existir Medidas de Control, sin Normas asociadas.
10. No pueden existir Normas sin Medidas de Control asociadas.
11. Pueden haber Tipos sin Medidas de control asociadas.
 - (a) **Obs:** No habría Medidas de Control para todo Tipo, con lo cual no existirían Productos con Medidas de control de todos los tipos.
12. Un Organismo puede controlar varios Tipos de Medidas de Control.
13. Un Tipo de Medida de Control es controlado por un solo Organismo.
14. Pueden existir Organismos sin Tipos de Medidas de Control asociados.
15. Todo Tipo de Medida de Control tiene un Organismo que lo controla.
16. Decidimos no modelar la parte de auditoría del problema, consideramos que no pertenecía directamente al mismo y que era mejor no incluirlo en el DER.

MR

pais

idpais
nombre

localidad

idlocalidad
nombre
idprovincia

rubro

idrubro
nombre

organismo

idorganismo
nombre

udemedida

umedida

medida

idmedida
idnorma
idtipo
tipo
desde
hasta

fabrica

idfabrica
cantempleados
idlocalidad

provincia

idprovincia
nombre

tipo

idtipo
nombre
idorganismo

norma

idnorma

producto

idproducto
nombre
descripcion
umedida
origen
idrubro

prod_extranjero-pais

idproducto
idpais
cantannual

prod_nacional_fabrica

idproducto
idfabrica

producto_extranjero

idproducto

producto_nacional

idproducto
cantpuestos

medida-producto

idmedida
idproducto

medida_por_prod_pais

idmedida

medida_por_producto

idmedida

extranjero_pais-medpp

idproducto
idmedida
idpais

usuario

nombreusuario

auditoria

idauditoria
usuario
descripcion
fecha_alteracion
idmedida_nueva
idmedida_vieja
idnorma_nueva
idnorma_vieja
idtipo_nueva
idtipo_vieja
tipo_nuevo
tipo_viejo
desde_nuevo
desde_viejo
hasta_nuevo
hasta_viejo

SQL

Stored Procedures

```
CREATE PROCEDURE `mydb`.`prodMasMC`()
BEGIN
SELECT  p.nombre
FROM
producto p,
(SELECT idproducto, count(*) as cantidad
FROM
((SELECT idproducto,idmedida FROM `medida-producto`) UNION ~
~ALL (SELECT idproducto,idmedida FROM `extranjero_pais-~
~medpp`)) a, medida c
WHERE a.idmedida = c.idmedida AND CURRENT_TIMESTAMP ~
~between c.desde and c.hasta
GROUP BY idproducto) b,
(SELECT MAX(cantidad) maximo FROM
(SELECT idproducto, count(*) as cantidad
FROM
((SELECT idproducto,idmedida FROM `medida-producto`) UNION~
~ ALL (SELECT idproducto,idmedida FROM `~
~extranjero_pais-medpp`)) a, medida c
WHERE a.idmedida = c.idmedida AND CURRENT_TIMESTAMP ~
~between c.desde and c.hasta
GROUP BY idproducto) b) n
WHERE p.idproducto = b.idproducto and n.maximo=b.cantidad;
END

CREATE PROCEDURE `mydb`.`prodTodasMC`()
BEGIN
SELECT  p.nombre FROM
producto p,
(SELECT DISTINCT idproducto, idtipo
FROM
((SELECT idproducto,idmedida FROM `medida-producto`) UNION ~
~ALL (SELECT idproducto,idmedida FROM `extranjero_pais-~
~medpp`)) a, medida c
WHERE a.idmedida = c.idmedida AND CURRENT_TIMESTAMP between~
~ c.desde and c.hasta
) b
WHERE p.idproducto = b.idproducto
GROUP BY b.idproducto
HAVING COUNT(*) = (select COUNT(idtipo) from tipo);
END
```

```

CREATE PROCEDURE `mydb`.`medidasPorRubro`()
BEGIN
    SELECT r.idrubro, r.nombre, medidasxrubro.cant_medidas
    FROM rubro r
    LEFT JOIN (
        SELECT p.rubro_idrubro, count(*) cant_medidas
        FROM producto p
        JOIN `medida-producto` m
        ON m.idproducto = p.idproducto
        GROUP BY p.rubro_idrubro
    ) AS medidasxrubro
    ON r.idrubro = medidasxrubro.rubro_idrubro;
END

```

Triggers

```

DROP TABLE IF EXISTS `mydb`.`auditoria`;

```

```

CREATE TABLE `mydb`.`auditoria`
(
    id int NOT NULL AUTO_INCREMENT,
    idmedida int,
    descripcion varchar(50),
    valor_anterior varchar(50),
    valor_nuevo varchar(50),
    usuario varchar(40),
    modificado datetime,
    PRIMARY KEY(id)
) ENGINE = InnoDB;

```

```

delimiter $$

```

```

DROP TRIGGER IF EXISTS medidaUpdate$$
CREATE TRIGGER medidaUpdate after UPDATE ON medida
FOR each row
BEGIN
    INSERT INTO auditoria (idmedida, descripcion, ~~
        ~~valor_anterior , valor_nuevo, usuario , modificado )~~
        ~~ VALUES(OLD.idmedida,'UPDATED' , OLD.idnorma , NEW.~~
        ~~idnorma, CURRENT_USER(), NOW() );
END;$$

```

```

DROP TRIGGER IF EXISTS medidaDelete$$

```

```

CREATE TRIGGER medidaDelete after DELETE ON medida
FOR each row
BEGIN
    INSERT INTO auditoria (idmedida, descripcion, ~
        ~valor_anterior , usuario , modificado ) VALUES(OLD.~
        ~idmedida,'DELETED' ,OLD.idnorma , CURRENT_USER(), ~
        ~NOW() );
END;$$

DROP TRIGGER IF EXISTS medidaInsert$$
CREATE TRIGGER medidaInsert after INSERT ON medida
FOR each row
BEGIN
    INSERT INTO auditoria (idmedida, descripcion, valor_nuevo, ~
        ~usuario , modificado ) VALUES (NEW.idmedida,'INSERTED',~
        ~ NEW.idnorma, CURRENT_USER(), NOW() );
END;$$

```