

Universidad ORT Uruguay
Facultad de Ingeniería

Obligatorio

Entregado como requisito del curso de la materia Base de Datos 2.

Dario Santellanes – 204582

Gabriel Pereyra – 224435

Profesores encargados:

Gabriel Bentos

Valeria Emanuelli

2018

Abstract

Se modelará un sistema para el funcionamiento general del proceso de producción de papel por parte de una empresa papelería y la gestión de su personal.

Se relevaron los requerimientos en base a la información provista y se construyó una solución en base de datos del problema utilizando el motor de base de datos Oracle y el lenguaje procedural PL/SQL.

Índice

Abstract	2
Índice	3
Introducción	5
Diagramas	5
Análisis de la solución	7
Diseño general	7
Identificación y Clasificación de Restricciones	9
Tabla EMPLEADO	9
Tabla PROVEEDOR	9
Tabla CLIENTE	9
Tabla LOTEMADERA	10
Tabla MADERACHIP	10
Tabla COCCION	11
Tabla ENERGIA	12
Tabla PAPEL	12
Tabla VENTA	13
Comentarios generales	13
Datos de prueba restricciones	14
Tabla CLIENTE	14
Tabla PROVEEDOR	14
Tabla EMPLEADO	14
Tabla LOTEMADERA	14
Tabla MADERACHIP	15
Trigger CONTROL_CALIDAD_CHIPEO	15
Trigger CONTROL_CAPATAZ_CHIPEO	16
Trigger ACTUALIZAR_PESO_CHIPEO	16
Tabla COCCION	16
Trigger CONTROL_PESO_COCCION	16
Tabla ENERGIA	16
Tabla PAPEL	17
Trigger CONTROL_STOCK	17
Tabla VENTA	17
Trigger CONTROL_VENTA	17

Requerimientos	18
Requerimiento 1 - Madera almacenada	18
Requerimiento 2 - Madera en mal estado	19
Requerimiento 3 - Control de calidad	20
Requerimiento 4 - Generación de energía	21
Requerimiento 5 - Producción de papel	22
Requerimiento 6 - Resumen de ventas	23
Requerimiento 7 - Resumen de clientes	25

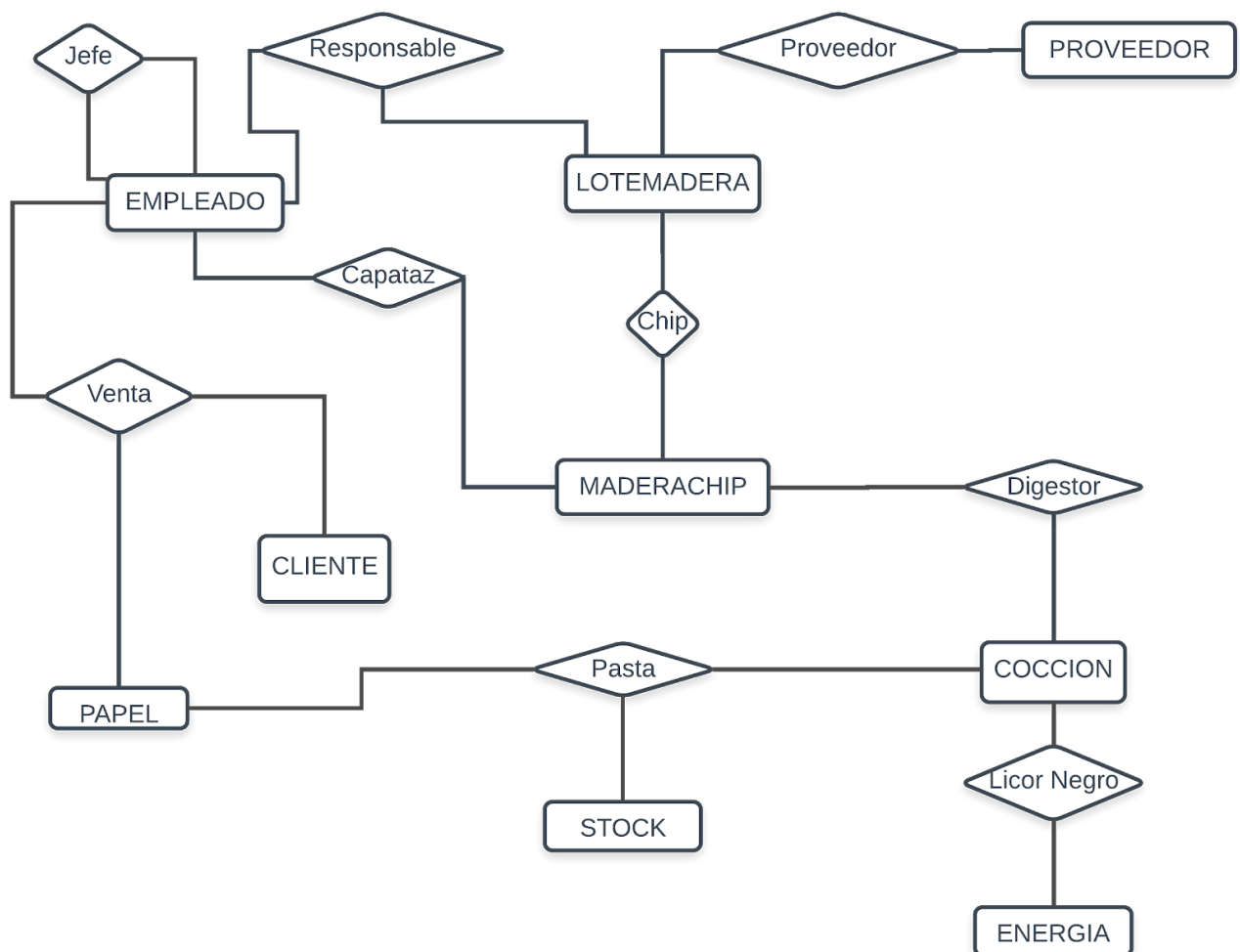
Introducción

El presente documento contiene la solución al problema presentado incluyendo, el diagrama del Modelo Entidad Relación, el DDL completo de la solución así como el análisis del diseño efectuado, los supuestos realizados y un análisis de las restricciones de integridad detectadas. También se incluyen los casos de prueba con las salidas esperadas por pantalla.

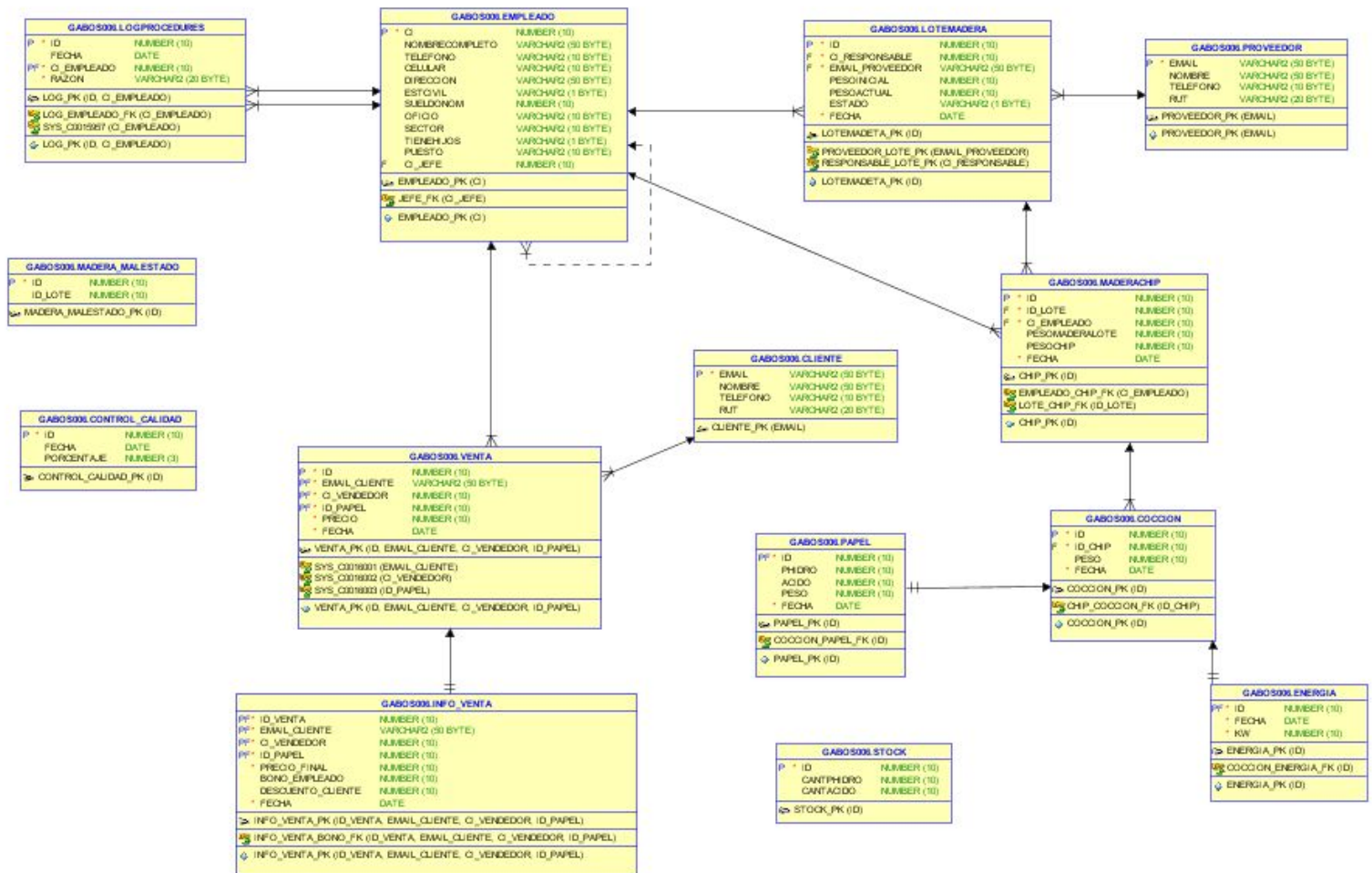
Diagramas

A modo ilustrativo se muestra un bosquejo del diagrama MER que se utilizó como guía para el desarrollo de la solución.

Es simplemente una guía por lo cual la solución final no se adapta completamente al diagrama ilustrado.



A continuación se muestra el diagrama generado desde la base de datos. El mismo se entrega adjunto en los archivos de entrega para su mejor visualización.



Análisis de la solución

El objetivo de esta sección del documento es el análisis de la solución propuesta y la justificación del diseño escogido.

Diseño general

La entidad **EMPLEADO** representa a los empleados de la empresa papelera junto con los sus datos personales. También se agrega una referencia a la misma entidad para representar el jefe del Empleado. En caso de ser jefe, el campo de referencia se mantiene sin valor.

La entidad **PROVEEDOR** representa a los proveedores de madera de la Papelera junto con los datos necesarios para representarlos.

La entidad **CLIENTE** representa a los clientes que tiene la empresa Papelera junto con los datos necesarios para representarlos.

La entidad **LOTEMADERA** representa los lotes de troncos de madera que se encuentran en el depósito. Se mantiene referencias al proveedor y al empleado responsable de recibir el lote de madera.

Se tiene un atributo fecha que indica la fecha de llegada del lote de madera, peso inicial que indica el peso inicial del lote y peso actual que indica el peso actual del lote ya que el lote se puede enviar a la chipeadora de a partes y no en su totalidad en una sola vez. Cuando se haya chipeado todo el lote, este campo quedará en cero.

La entidad **MADERACHIP** representa los chips generados luego de ingresar troncos de madera a la Chipeadora. Se mantiene una referencia al empleado encargado del chipeo así como también el peso de madera del lote que ingresa, el peso en chips resultado y un atributo fecha para saber la fecha en la que se realizó el chipeo.

La entidad **COCCION** representa el envío y procesamiento de los chips de madera en el Digestor. Se tienen los atributos peso que representa el peso de los chips que ingresan al digestor, la fecha en la que se realiza la operación y una referencia al registro de chips que se ingresan al Digestor.

La entidad **ENERGIA** representa el licor negro y posterior generación de energía con ello que se realiza dentro del proceso de producción. Se tienen los atributos de fecha en la que se realiza el proceso y la cantidad de kw que se generan.

La entidad **PAPEL** representa al papel generado mediante la pasta resultante luego de la cocción en el digestor. Se tienen los atributos de fecha que representa la fecha en la que se realiza el proceso, el peso de papel generado y la cantidad de hidrógeno y ácidos necesarios para la producción.

Los identificadores de las tablas PAPEL y ENERGIA son referencias a la tabla COCCION ya que son el resultado de la cocción en el Digestor.

La entidad **STOCK** representa el stock de ácidos e hidrógeno que se tienen en la papelera para utilizar en la producción del papel.

La entidad **VENTA** representa las ventas de papel generado que se realizan. Se tienen referencias al vendedor que las realiza, el cliente que realiza la compra del papel y el papel que se está vendiendo. También se tiene un atributo de fecha en la que se realiza la venta y el precio de la misma.

También existen otras tablas que son auxiliares para el cumplimiento de los restantes requerimientos solicitados. Estas tablas se explicarán más adelante su utilidad.

Supuestos efectuados

- Por el tipo de solución optado la tabla de stock siempre va a mantener un solo registro con el stock existente y el mismo se va a ir actualizando cuando se da de alta registros en PAPEL.
- Una vez chipeada la madera, entra al digestor en su totalidad.
- No pueden hacerse más de una cocción sobre la misma chipeada.
- En el resumen de ventas no se muestra el bono mensual que recibe el empleado ya que ese bono no existe por una venta en especial sino por una sumatoria de ventas en el mes.
- En el requerimiento seis (Resumen de ventas) el mes recibido por parámetro es para el año en curso. Ya que sino debería tener que recibir el año por parámetro también.

Identificación y Clasificación de Restricciones

Tabla EMPLEADO

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
CI NOT NULL PRIMARY KEY	Entidad	Estructural
CIJEFE DEFAULT NULL	Dominio	Estructural
CIJEFE CONSTRAINT jefe_fk REFERENCES EMPLEADO	Referencial	Estructural
CHECK(ESTCIVIL IN ('S','C'))	Dominio	Estructural
CHECK(TIENEHIJOS IN ('S','N'))	Dominio	Estructural

Tabla PROVEEDOR

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
EMAIL NOT NULL PRIMARY KEY	Entidad	Estructural
RUT DEFAULT NULL	Dominio	Estructural

Tabla CLIENTE

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
EMAIL NOT NULL PRIMARY KEY	Entidad	Estructural

Tabla LOTEMADERA

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
ID NOT NULL PRIMARY KEY	Entidad	Estructural
CI_RESPONSABLE NOT NULL	Dominio	Estructural
EMAIL_PROVEEDOR NOT NULL	Dominio	Estructural
CI_RESPONSABLE CONSTRAINT responsable_lote_pk REFERENCES EMPLEADO	Referencial	Estructural
EMAIL_PROVEEDOR CONSTRAINT proveedor_lote_pk REFERENCES PROVEEDOR	Referencial	Estructural
CHECK(PESOINICIAL > 0)	Dominio	Estructural
CHECK(PESOACTUAL >= 0)	Dominio	Estructural
CHECK (ESTADO IN ('S', 'N'))	Dominio	Estructural
LOTEMADERA_ID	Semántica	No Estructural

Tabla MADERACHIP

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
ID NOT NULL PRIMARY KEY	Entidad	Estructural
ID_LOTE NOT NULL	Dominio	Estructural
CI_EMPLEADO NOT NULL	Dominio	Estructural
FECHA NOT NULL	Dominio	Estructural
ID_LOTE CONSTRAINT lote_chip_fk REFERENCES LOTEMADERA	Referencial	Estructural
CI_EMPLEADO CONSTRAINT empleado_chip_fk REFERENCES EMPLEADO	Referencial	Estructural
CHECK(PESOMADERALOTE > 0)	Dominio	Estructural
CHECK(PESOCHIP >= 0)	Dominio	Estructural
CONTROL_CAPATAZ_CHIPEO	Semántica	No Estructural

CONTROL_CALIDAD_CHIPEO	Semántica	No Estructural
MADERACHIP_ID	Semántica	No Estructural

Comentarios:

- El trigger **CONTROL_CAPATAZ_CHIPEO** controla que el empleado responsable de llevar un lote de madera a chipear sea un jefe.
- El trigger **CONTROL_CALIDAD_CHIPEO** realiza el requerimiento tres solicitado en donde controla que el peso de la madera ingresada a la chiheadora y el peso de la madera chip resultante exista como máximo un 5% por día.
- Se crea una vista sobre la tabla MADERACHIP debido a error de mutating al ejecutarse el trigger **ACTUALIZAR_PESO_CHIPEO**

La vista se genera de la siguiente manera:

```
CREATE VIEW VIEW_MADERACHIP AS
SELECT ID,ID_LOTE,CI_EMPLEADO,PESOMADERALOTE,PESOCHIP,FECHA
FROM MADERACHIP;
```

Luego el trigger **ACTUALIZAR_PESO_CHIPEO** lo que realiza es actualizar el campo PESOACTUAL del lote de madera correspondiente al momento de dar de alta un registro en MADERACHIP. Esto se debe a que un lote de madera puede enviarse a la chiheadora en partes y no todo de una sola vez.

Tabla COCCION

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
ID NOT NULL PRIMARY KEY	Entidad	Estructural
ID_CHIP NOT NULL	Dominio	Estructural
FECHA NOT NULL	Dominio	Estructural
CHECK(PESO >= 0)	Dominio	Estructural
ID_CHIP CONSTRAINT chip_coccion_fk REFERENCES MADERACHIP	Referencial	Estructural
CONTROL_PESO_COCCION	Semántica	No Estructural
COCCION_ID	Semántica	No Estructural

Comentarios:

- El trigger **CONTROL_PESO_COCCION** controla que el peso de la cocción no sea superior que el peso de los chips ingresados al digestor

Tabla ENERGIA

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
ID NOT NULL PRIMARY KEY	Entidad	Estructural
KW NOT NULL	Dominio	Estructural
FECHA NOT NULL	Dominio	Estructural
ID CONSTRAINT coccion_energia_fk REFERENCES COCCION	Referencial	Estructural

Tabla PAPEL

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
ID NOT NULL PRIMARY KEY	Entidad	Estructural
FECHA NOT NULL	Dominio	Estructural
CHECK(PESO > 0)	Dominio	Estructural
CHECK(PHIDRO > 0)	Dominio	Estructural
CHECK(ACIDO > 0)	Dominio	Estructural
ID CONSTRAINT coccion_papel_fk REFERENCES COCCION	Referencial	Estructural
CONTROL_STOCK	Semántica	No Estructural

Comentarios:

- El trigger **CONTROL_STOCK** controla que exista stock de la cantidad de hidrógeno y ácidos utilizados para la generación del papel

Tabla VENTA

RESTRICCIÓN	TIPO RESTRICCIÓN	IMPLEMENTACIÓN
ID,EMAIL_CLIENTE,CI_VENDEDOR,ID_PAPEL PRIMARY KEY	Entidad	Estructural
EMAIL_CLIENTE REFERENCES CLIENTE	Referencial	Estructural
CI_VENDEDOR REFERENCES EMPLEADO	Referencial	Estructural
ID_PAPEL REFERENCES PAPEL	Referencial	Estructural
FECHA NOT NULL	Dominio	Estructural
PRECIO NOT NULL	Dominio	Estructural
CONTROL_VENTA	Semántica	No Estructural
VENTA_ID	Semántica	No Estructural

Comentarios:

- El trigger **CONTROL_VENTA** realiza los descuentos en la venta ya sea porque supera el precio la venta individual o hay que hacerle un descuento extra porque el cliente supera el importe de compras mensuales.

Comentarios generales

- Para el manejo de ids autoincrementales en las tablas se definen sequences que permiten extraer valores numéricos e insertarlos, incrementalmente, en los ids correspondientes. Para ello también se utilizan triggers, con la cláusula before insert, en las tablas que tengan ids como clave primaria y, previo a insertar un nuevo registro, este valor se incrementa. Este es el caso de por ejemplo las tablas LOTEMADERA, MADERACHIP, COCCION y VENTA que definen los triggers LOTEMADERA_ID, MADERACHIP_ID, COCCION_ID y VENTA_ID respectivamente que incrementan el valor de las sequences definidas.
- Existen tablas auxiliares que se explicarán más adelante su utilidad pero que no se mencionan sus restricciones por ser tablas auxiliares.

Datos de prueba restricciones

Tabla CLIENTE

Si se quiere ingresar un cliente con el email vacío obtenemos error como se desea.

Entrada:

```
INSERT INTO CLIENTE(EMAIL,RUT,NOMBRE,TELEFONO)
VALUES ('','213456740012','ABS S.R.L','25678903')
```

Salida:

Informe de error - ORA-01400: cannot insert NULL into ("CLIENTE"."EMAIL")

Tabla PROVEEDOR

Si se quiere ingresar un proveedor con email vacío obtenemos error como se desea.

Entrada:

```
INSERT INTO PROVEEDOR(EMAIL,NOMBRE,RUT,TELEFONO)
VALUES ('','PROV1','213456789014','23096745');
```

Salida:

Informe de error - ORA-01400: cannot insert NULL into ("PROVEEDOR"."EMAIL")

Tabla EMPLEADO

Si se quiere ingresar un empleado con cédula vacía obtenemos un error análogo a los anteriores

También se obtienen error y no deja insertar registros si los campos ESTCIVIL y TINEHIJOS tienen un valor distinto al permitido en los CHECK

Tabla LOTEMADERA

Se obtienen errores si el peso del lote no es mayor que 0 o si el estado no esta dentro de los valores posibles.

Tambien se genera error si no se pone un email de un empleado existente como por ejemplo:

Entrada:

INSERT INTO

LOTEMADERA(EMAIL_PROVEEDOR,PESOINICIAL,ESTADO,FECHA,PESOACTUAL,CI_RESPONSABLE)
VALUES ('prov1@gmail.com',1000,'S','20/06/2018',1000,12345678);

Salida:

Informe de error -

ORA-02291: integrity constraint (PROVEEDOR_LOTE_PK) violated - parent key not found

Lo mismo sucede si se ingresa una cédula de un empleado que no existe.

Tabla MADERACHIP

Los controles de NOT NULL, CHECK y las REFERENCES son similares a los mencionados anteriormente pero para los campos mencionados en la sección de restricciones.

Al momento del testeo surgió un error de mutating en un trigger, por lo cual se tuvo que buscar una solución. La misma fue crear una vista sobre MADERACHIP y modificar el trigger de ACTUALIZAR_PESO_CHIPEO para que sea sobre la vista.

Trigger CONTROL_CALIDAD_CHIPEO

Dicho trigger controla la restricción de que en el día la diferencia de peso que entró a la chipeadora y el peso de la madera chipeada no supere el 5%.

También se controla que el peso que se ingrese a la chipeadora no sea mayor que el peso actual del lote que se referencia.

A su vez en este trigger se utiliza una tabla auxiliar CONTROL_CALIDAD en la cual se va agregando o eliminando un registro para la fecha que se procesa si en la totalidad se está superando o no la restricción del 5%.

Un ejemplo sería:

Entrada:

INSERT INTO

VIEW_MADERACHIP(ID_LOTE,CI_EMPLEADO,PESOMADERALOTE,PESOCHIP,FECHA)
VALUES(1,12345678,650,550,'20/06/2018');

Salida:

Se da de alta el registro en MADERACHIP.

Se da de alta en la tabla auxiliar CONTROL_CALIDAD un registro con la fecha y el porcentaje de chipeo.

ID	FECHA	PORCENTAJE
1	24-06-2018 18:15:44	85

Trigger CONTROL_CAPATAZ_CHIPEO

Este trigger controla que el empleado responsable de chippear sea un jefe, si esto no sucede se obtiene error como el siguiente:

Entrada:

```
INSERT INTO  
VIEW_MADERACHIP(ID_LOTE,CI_EMPLEADO,PESOMADERALOTE,PESOCHIP,FECHA)  
VALUES(1,23456789,650,630,'20/06/2018');
```

Salida:

Informe de error - ORA-20003: Los chips deben ser llevados por capataces

Trigger ACTUALIZAR_PESO_CHIPEO

ID	CI_RESPONSABLE	EMAIL_PROVEEDOR	PESOINICIAL	PESOACTUAL	ESTADO	FECHA
1	12345678	prov1@gmail.com	1000	350	S	20-06-2018 00:00:00
2	34567890	prov2@gmail.com	10000	10000	S	30-06-2017 00:00:00
3	12345678	prov3@gmail.com	2000	2000	S	24-04-2018 00:00:00
4	34567890	prov4@gmail.com	5000	5000	S	17-03-2017 00:00:00
5	12345678	prov1@gmail.com	2500	2500	S	08-10-2017 00:00:00

Con el mismo ejemplo del trigger CONTROL_CALIDAD_CHIPEO, se puede ver en la imagen como se le restó al registro de id uno los 650 del registro de madera chip en el campo PESOACTUAL. Ya que inicialmente PESOACTUAL se da de alta con el mismo valor que PESOINICIAL.

Tabla COCCION

Trigger CONTROL_PESO_COCCION

Controla si el peso de la cocción no supere al peso de los chip asociados.

Entrada:

```
INSERT INTO COCCION(ID_CHIP,FECHA,PESO) VALUES(2,'20/06/2018',630)
```

Salida:

Informe de error - ORA-20002: Fallo el control de peso

Tabla ENERGIA

Los controles en esta tabla son los de NOT NULL y REFERENCES análogos a los ya mencionados para las tablas anteriores.

Tabla PAPEL

Esta tabla tiene los controles de NOT NULL, REFERENCES y CHECK como para tablas anteriores.

Trigger CONTROL_STOCK

Este trigger quita de la tabla de stock el hidrógeno y ácido necesario para la elaboración del papel, disminuyendo la cantidad que existente en la tabla de stock.

Tabla VENTA

Esta tabla tiene los controles de NOT NULL, REFERENCES y CHECK como para tablas anteriores.

Trigger CONTROL_VENTA

Este trigger realiza los descuentos mencionados en la letra del obligatorio si se cumplen las condiciones que se mencionan. Estos descuentos se agregan en una tabla auxiliar INFO_VENTAS que luego va a ser utilizada por los requerimientos para mantener la información resumida en dicha tabla.

Requerimientos

Requerimiento 1 - Madera almacenada

Para este requerimiento se realiza un procedure que recorre los lotes de madera ordenado ascendente por fecha para lograr tener los más próximos a vencer primeros. Luego de esto se imprimen los valores solicitados.

```
-- REQUERIMIENTO MADERA ALMACENADA
CREATE OR REPLACE PROCEDURE MADERA_ALMACENADA AS
BEGIN
    DECLARE
        CURSOR MADERA IS
            SELECT L.ID, L.FECHA, L.PESOACTUAL, L.EMAIL_PROVEEDOR FROM LOTEMADERA L
            WHERE L.ESTADO = 'S'
            ORDER BY L.FECHA ASC;
    BEGIN
        DBMS_OUTPUT.PUT_LINE('ID LOTE' || ' ' || 'FECHA LOTE' || ' ' || 'PESO ACTUAL' || ' ' || 'EMAIL PROV');
        FOR itemLote IN MADERA
        LOOP
            DBMS_OUTPUT.PUT_LINE(rpad(itemLote.ID,12) || rpad(itemLote.FECHA,15) || rpad(itemLote.PESOACTUAL,16) || rpad(itemLote.EMAIL_PROVEEDOR,20));
        END LOOP;
    END;
END;
```

La forma de ejecutarlo es: EXECUTE MADERA_ALMACENADA()

Un ejemplo de salida que se puede ver en la consola es:

ID LOTE	FECHA LOTE	PESO ACTUAL	EMAIL PROV
4	17-03-2017	2000	prov4@gmail.com
2	30-06-2017	5000	prov2@gmail.com
5	08-10-2017	2500	prov1@gmail.com
3	24-04-2018	100	prov3@gmail.com
1	20-06-2018	0	prov1@gmail.com

Requerimiento 2 - Madera en mal estado

Para este requerimiento se realizó un procedure que actualiza el estado de los lotes de madera, cambiando el estado a los lotes que ya tienen más de seis meses en el depósito.

Se utiliza en este procedure la tabla auxiliar MADERA_MALESTADO en la cual se va guardando el id del lote que se está procesando, ya que si el proceso se interrumpe luego se puede continuar a partir de dicho registro (leyendo en esta tabla) sin tener que recorrer todos los anteriores.

```
-- REQUERIMIENTO MADERA EN MAL ESTADO
CREATE OR REPLACE PROCEDURE MADERA_MAL_ESTADO(FECHA_ENTRADA IN DATE) AS
BEGIN
    DECLARE
        ULTREGISTRO NUMBER(10);
        FECHALIMITE DATE;

        -- ME GUARDO TODOS LOS LOTES QUE ESTAN EN BUEN ESTADO
        CURSOR MADERA IS
            SELECT L.ID, L.FECHA FROM LOTEMADERA L
            WHERE L.ESTADO = 'S';
    BEGIN
        FECHALIMITE := ADD_MONTHS(FECHA_ENTRADA, -6);
        SELECT NVL(MAX(ID_LOTE), 0) INTO ULTREGISTRO FROM MADERA_MALESTADO;

        FOR itemLote IN MADERA
        LOOP
            UPDATE LOTEMADERA SET ESTADO = 'N' WHERE FECHA < FECHALIMITE AND ID >= ULTREGISTRO;
            INSERT INTO MADERA_MALESTADO(ID_LOTE) VALUES (itemLote.ID);
        END LOOP;

        EXECUTE IMMEDIATE 'TRUNCATE TABLE MADERA_MALESTADO';
        COMMIT;
    END;
END;
```

Un ejemplo de ejecución sería: EXECUTE MADERA_MAL_ESTADO(CURRENT_DATE);

Antes de la ejecución la tabla LOTEMADERA es:

ID	CI_RESPONSABLE	EMAIL_PROVEEDOR	PESOINICIAL	PESOACTUAL	ESTADO	FECHA
1	12345678	prov1@gmail.com	1000	0	S	20-06-2018
2	34567890	prov2@gmail.com	10000	5000	S	30-06-2017
3	12345678	prov3@gmail.com	2000	100	S	24-04-2018
4	34567890	prov4@gmail.com	5000	2000	S	17-03-2017
5	12345678	prov1@gmail.com	2500	2500	S	08-10-2017

Luego de la ejecución la tabla LOTEMADERA es:

ID	CI_RESPONSABLE	EMAIL_PROVEEDOR	PESOINICIAL	PESOACTUAL	ESTADO	FECHA
1	12345678	prov1@gmail.com	1000	0	S	20-06-2018
2	34567890	prov2@gmail.com	10000	5000	N	30-06-2017
3	12345678	prov3@gmail.com	2000	100	S	24-04-2018
4	34567890	prov4@gmail.com	5000	2000	N	17-03-2017
5	12345678	prov1@gmail.com	2500	2500	N	08-10-2017

Requerimiento 3 - Control de calidad

Este requerimiento se realizó mediante un trigger en la tabla MADERACHIP.

En un principio se había realizado que el trigger devolviera un error cuando no se cumple con el requerimiento pero como el control tiene que ser por día se debe hacer que en una tabla auxiliar se vaya guardando para que día no se cumple el requerimiento.

Es decir que el trigger si se cumple el requerimiento, busca en la tabla auxiliar si existe un registro y lo elimina. Si no se cumple el requerimiento, entonces se agrega un registro o se actualiza el porcentaje en caso de que ya existiera el registro.

El código del trigger es el siguiente:

```
CREATE OR REPLACE TRIGGER CONTROL_CALIDAD_CHIPEO BEFORE INSERT OR UPDATE ON MADERACHIP
FOR EACH ROW
--FOLLOWS CONTROL_PESO_CHIPEO
DECLARE
    v_peso_chip NUMBER(10);
    v_peso_madera NUMBER(10);
    v_porcentaje NUMBER(10);
    v_control_id NUMBER(10);
    v_peso_lote NUMBER(10);
BEGIN
    -- CONTROL EL PESO DEL LOTE
    SELECT 1.PESOACTUAL INTO v_peso_lote FROM LOTEMADERA 1 WHERE :NEW.ID_LOTE = 1.ID;

    IF UPDATING THEN
        v_peso_lote := v_peso_lote + :OLD.PESOMADERALOTE;
    END IF;

    IF (v_peso_lote < :NEW.PESOMADERALOTE) THEN
        Raise_Application_Error (-20002, 'Fallo el control de peso');
    END IF;

    -- CONTROL DE CALIDAD DEL 95%
    SELECT NVL(SUM(m.PESOCCHIP),0) INTO v_peso_chip FROM MADERACHIP m WHERE TRUNC(m.FECHA) = TRUNC(CURRENT_DATE);
    SELECT NVL(SUM(m.PESOMADERALOTE),0) INTO v_peso_madera FROM MADERACHIP m WHERE TRUNC(m.FECHA) = TRUNC(CURRENT_DATE);

    IF UPDATING THEN
        v_peso_chip := v_peso_chip - :OLD.PESOCCHIP;
        v_peso_madera := v_peso_madera - :OLD.PESOMADERALOTE;
    END IF;

    v_peso_chip := v_peso_chip + :NEW.PESOCCHIP;
    v_peso_madera := v_peso_madera + :NEW.PESOMADERALOTE;
    v_porcentaje := v_peso_chip * 100 / v_peso_madera;

    -- SI ESTOY POR DEBAJO DEL CONTROL DE CALIDAD AGREGO REGISTRO O MODIFICO EXISTENTE, SI NO ELIMINO EL EXISTENTE.
    IF (v_porcentaje < 95) THEN
        -- BUSCO REGISTRO PARA SABER SI DAR DE ALTA O MODIFICAR
        SELECT COUNT(*) INTO v_control_id FROM CONTROL_CALIDAD WHERE TRUNC(FECHA) = TRUNC(SYSDATE);

        IF v_control_id = 0 THEN
            INSERT INTO CONTROL_CALIDAD(FECHA,PORCENTAJE) VALUES (CURRENT_DATE,v_porcentaje);
        ELSE
            UPDATE CONTROL_CALIDAD SET PORCENTAJE = v_porcentaje WHERE TRUNC(FECHA) = TRUNC(SYSDATE);
        END IF;
    ELSE
        DELETE FROM CONTROL_CALIDAD WHERE TRUNC(FECHA) = TRUNC(SYSDATE);
    END IF;
END;
```

Requerimiento 4 - Generación de energía

Para este requerimiento se realizó un procedure que imprime en consola los resultados solicitados.

Dicho procedimiento recibe por parámetro una fecha desde y otra fecha hasta, las cuales se utilizan para filtrar los datos para ese período de tiempo.

El código del mismo es:

```
-- REQUERIMIENTO GENERACION DE ENERGIA
CREATE OR REPLACE PROCEDURE GENERACION_ENERGIA(FECHA_DESDE IN DATE, FECHA_HASTA IN DATE) AS
BEGIN
    DECLARE
        cant_madera_lote NUMBER(10);
        cant_madera_chip NUMBER(10);

        -- AGRUPO POR FECHA
        CURSOR ENERGIA IS
            SELECT E.FECHA AS FECHA, SUM(KW) AS KW
            FROM ENERGIA E
            WHERE E.FECHA >= FECHA_DESDE AND E.FECHA <= FECHA_HASTA
            GROUP BY E.FECHA
            ORDER BY E.FECHA;

    BEGIN
        DBMS_OUTPUT.PUT_LINE('FECHA' || ' ' || 'CANT. KW' || ' ' || 'PESO LOTE' || ' ' || 'PESO CHIP');

        FOR itemEnergia IN ENERGIA
        LOOP
            SELECT SUM(M.PESOMADERALOTE) INTO cant_madera_lote
            FROM MADERACHIP M
            WHERE M.FECHA = itemEnergia.FECHA;

            SELECT SUM(C.PESO) INTO cant_madera_chip
            FROM COCCION C
            WHERE C.FECHA = itemEnergia.FECHA;

            DBMS_OUTPUT.PUT_LINE(rpad(itemEnergia.FECHA,12) || rpad(itemEnergia.KW,12) || rpad(cant_madera_lote,16) || rpad(cant_madera_chip,12));
        END LOOP;
    END;
END;
```

La salida a la ejecución: EXECUTE GENERACION_ENERGIA('18/06/2018','23/06/2018');

FECHA	CANT. KW	PESO LOTE	PESO CHIP
20-06-2018	1660	1000	830
21-06-2018	13770	6900	6885
22-06-2018	4400	3000	2200

Requerimiento 5 - Producción de papel

Para este requerimiento se realizó un procedure que imprime en consola los resultados solicitados.

Dicho procedimiento recibe por parámetro una fecha desde y otra fecha hasta, las cuales se utilizan para filtrar los datos para ese período de tiempo.

El código del mismo es:

```
-- REQUERIMIENTO PRODUCCION DE PAPEL
CREATE OR REPLACE PROCEDURE PRODUCCION_PAPEL(FECHA_DESDE IN DATE, FECHA_HASTA IN DATE) AS
BEGIN
    DECLARE
        v_fecha DATE;
        v_sum_mchip NUMBER(10);
        v_sum_lmadera NUMBER(10);
        -- AGRUPO POR FECHA
        CURSOR GRUPO IS
            SELECT SUM(p.PESO) AS SUMA, p.FECHA as FECHA FROM PAPEL p WHERE p.FECHA >= FECHA_DESDE AND p.FECHA <= FECHA_HASTA
            GROUP BY p.FECHA
            ORDER BY p.FECHA;
    BEGIN
        DBMS_OUTPUT.PUT_LINE('FECHA' || ' ' || 'PESO PAPEL' || ' ' || 'PESO CHIPS' || ' ' || 'PESO LOTE');

        FOR itemGrupo IN GRUPO
        LOOP
            SELECT NVL(SUM(m.PESOCCHIP),0) into v_sum_mchip FROM MADERACHIP m WHERE m.FECHA = itemGrupo.FECHA;
            SELECT NVL(SUM(l.PESOCINICIAL),0) into v_sum_lmadera FROM LOTEMADERA l WHERE l.FECHA = itemGrupo.FECHA;
            DBMS_OUTPUT.PUT_LINE(rpad(itemGrupo.FECHA,12) || rpad(itemGrupo.SUMA,14) || rpad(v_sum_mchip,14) || rpad(v_sum_lmadera,12));
        END LOOP;
    END;
END;
```

La salida a la ejecución: EXECUTE PRODUCCION_PAPEL('18/06/2018','23/06/2018');

FECHA	PESO PAPEL	PESO CHIPS	PESO LOTE
20-06-2018	415	830	1000
21-06-2018	3490	6885	0
22-06-2018	1100	2200	0

Requerimiento 6 - Resumen de ventas

Este requerimiento se resuelve mediante un procedure que imprime en consola los resultados como se piden en la letra del obligatorio.

Como se puso en los supuestos, el mes pasado por parámetro se toma como año el actual. En caso de ser vacío se toma el mes anterior al actual.

El código es el siguiente:

```
create or replace PROCEDURE RESUMEN_VENTAS(CI_EMPLEADO IN NUMBER, MES IN NUMBER DEFAULT 0) AS
BEGIN
    DECLARE
        v_fecha DATE;
        v_mes_actual NUMBER;

        CURSOR VENDEDORES IS
            SELECT e.CI, e.NOMBRECOMPLETO FROM (SELECT DISTINCT v.CI_VENDEDOR FROM VENTA v) lista, EMPLEADO e, VENTA v
            WHERE e.CI = lista.CI_VENDEDOR
            AND e.CI = v.CI_VENDEDOR
            AND EXTRACT(MONTH from v_fecha) = EXTRACT(MONTH from v.FECHA)
            AND EXTRACT(YEAR from v_fecha) = EXTRACT(YEAR from v.FECHA)
            ;

        v_vendedores_rec VENDEDORES%ROWTYPE;

        CURSOR INFO IS
            SELECT * FROM INFO_VENTA iv WHERE iv.CI_VENDEDOR = v_VENDEDORES_rec.CI
            AND EXTRACT(MONTH from iv.FECHA) = EXTRACT(MONTH from v_fecha);

    BEGIN

        if(MES = 0) THEN
            SELECT ADD_MONTHS(SYSDATE,-1) into v_fecha FROM dual;
        ELSE
            SELECT ADD_MONTHS(SYSDATE, (MES - EXTRACT(MONTH from SYSDATE))) into v_fecha FROM dual;
        END IF;

        /*****Crea Log*****/

        INSERT INTO LOGPROCEDURES(FECHA, CI_EMPLEADO, RAZON) VALUES(SYSDATE, CI_EMPLEADO, 'RESUMEN_VENTAS');

        OPEN VENDEDORES;

        LOOP
            FETCH VENDEDORES into v_vendedores_rec;
            EXIT WHEN VENDEDORES%NOTFOUND;

            DEMS_OUTPUT.PUT_LINE('Empleado: ' || rpad(v_vendedores_rec.NOMBRECOMPLETO,50));
            DEMS_OUTPUT.PUT_LINE('EMAIL CLIENTE' || ' ' || 'PRECIO FINAL' || ' ' || 'BONOS OBTENIDOS');

            FOR info_venta IN INFO
            LOOP
                DEMS_OUTPUT.PUT_LINE(rpad(info_venta.EMAIL_CLIENTE,25) || rpad(info_venta.PRECIO_FINAL,17) || rpad(info_venta.BONO_EMPLEADO,12));
            END LOOP;

            DEMS_OUTPUT.PUT_LINE(' ');
        END LOOP;
        CLOSE VENDEDORES;

    END;
END;
```

Un ejemplo de ejecución para el mes de Junio sería

EXECUTE RESUMEN_VENTAS(12345678,6);

Empleado: Rafael Rodriguez

EMAIL CLIENTE	PRECIO FINAL	BONOS OBTENIDOS
abc@gmail.com	10	0

Empleado: Pedro Gimenez

EMAIL CLIENTE	PRECIO FINAL	BONOS OBTENIDOS
abc@gmail.com	1380	30

Empleado: Jose Umpierrez

EMAIL CLIENTE	PRECIO FINAL	BONOS OBTENIDOS
nop@gmail.com	920	20

Requerimiento 7 - Resumen de clientes

Este requerimiento se resolvió mediante un procedure el cual recibe como parámetro el período a filtrar y cuando es vacío se debe considerar todo el año actual.

```
CREATE OR REPLACE PROCEDURE RESUMEN_CLIENTES
(
    CI_EMPLEADO IN NUMBER,
    FECHA_INICIO IN DATE DEFAULT TRUNC(SYSDATE, 'YEAR'),
    FECHA_FINAL IN DATE DEFAULT TRUNC(ADD_MONTHS(SYSDATE, 1))
) AS
BEGIN
    DECLARE
        v_mes_actual NUMBER;
        CURSOR CLIENTES IS
            SELECT C.Xmelli, c.Nombre FROM
            (
                SELECT v.EMAIL_CLIENTE FROM VENTA v, CLIENTES c
                WHERE v.EMAIL_CLIENTE = c.EMAIL
                AND EXTRACT(MONTH FROM v.FECHA) >= EXTRACT(MONTH FROM FECHA_INICIO)
                AND EXTRACT(MONTH FROM v.FECHA) <= EXTRACT(MONTH FROM FECHA_FINAL)
                AND EXTRACT(YEAR FROM v.FECHA) >= EXTRACT(YEAR FROM FECHA_INICIO)
                AND EXTRACT(YEAR FROM v.FECHA) <= EXTRACT(YEAR FROM FECHA_FINAL)
                GROUP BY v.EMAIL_CLIENTE
                ORDER BY SUM(v.PRECIO) DESC
            ) lista, CLIENTES c
            WHERE lista.Xmelli_Cliente = C.Xmelli
            AND ROWNUM <= 10;

        v_clientes_rec CLIENTES%ROWTYPE;
        CURSOR INFO IS
            SELECT SUM(s.PRECIO_FINAL) AS CORRIAS, SUM(i.DISCUENTO_CLIENTE) AS DISCUIENTOS FROM INFO_VENTA i
            WHERE i.EMAIL_CLIENTE = v_clientes_rec.Xmelli AND (EXTRACT(MONTH FROM i.FECHA) = v_mes_actual);

        v_info_rec INFO%ROWTYPE;
        BEGIN
            OPEN CLIENTES;

            LOOP
                FETCH CLIENTES INTO v_clientes_rec;
                EXIT WHEN CLIENTES%NOTFOUND;

                DBMS_OUTPUT.PUT_LINE('Clientes: ' || LPAD(v_clientes_rec.MEMBERS, 30));
                FOR mes IN (EXTRACT(MONTH FROM FECHA_INICIO))..(EXTRACT(MONTH FROM FECHA_FINAL)) LOOP
                    v_mes_actual := mes;
                    DBMS_OUTPUT.PUT_LINE('MES: ' || TO_CHAR(TO_DATE(v_mes_actual, 'PM'), 'MONTH'));
                    DBMS_OUTPUT.PUT_LINE('TOTAL CORRIAS: ' || ' ' || DISCUIENTOS);
                    OPEN INFO;
                    LOOP
                        FETCH INFO INTO v_info_rec;
                        EXIT WHEN INFO%NOTFOUND;

                        DBMS_OUTPUT.PUT_LINE(LPAD(v_info_rec.CORRIAS, 15) || LPAD(v_info_rec.DISCUIENTOS, 15));
                        DBMS_OUTPUT.PUT_LINE(' ');
                    END LOOP;
                    CLOSE INFO;
                END LOOP;
                DBMS_OUTPUT.PUT_LINE('-----');
            END LOOP;
            CLOSE CLIENTES;

            INSERT INTO LOGPROCEDURES(FECHA, CI_EMPLEADO, RAZON) VALUES (SYSDATE, CI_EMPLEADO, 'RESUMEN_CLIENTES');
            COMMIT;
        END;
    END;
```

El resultado de la ejecución de:
EXECUTE RESUMEN_CLIENTES(12345678,'10/02/2018','10/09/2018');

<p>Cliente: NOP S.R.L</p> <p>MES: FEBRERO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: SEPTIEMBRE</p> <p>TOTAL COMPRAS DESCUENTOS</p>
<p>MES: MARZO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>-----</p> <p>Cliente: ABS S.R.L</p> <p>MES: FEBRERO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>-----</p> <p>Cliente: DEF S.R.L</p> <p>MES: FEBRERO</p> <p>TOTAL COMPRAS DESCUENTOS</p>
<p>MES: ABRIL</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: MARZO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: MARZO</p> <p>TOTAL COMPRAS DESCUENTOS</p>
<p>MES: MAYO</p> <p>TOTAL COMPRAS DESCUENTOS</p> <p>1840 160</p>	<p>MES: ABRIL</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: ABRIL</p> <p>TOTAL COMPRAS DESCUENTOS</p> <p>900 0</p>
<p>MES: JUNIO</p> <p>TOTAL COMPRAS DESCUENTOS</p> <p>920 80</p>	<p>MES: MAYO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: MAYO</p> <p>TOTAL COMPRAS DESCUENTOS</p> <p>200 0</p>
<p>MES: JULIO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: JUNIO</p> <p>TOTAL COMPRAS DESCUENTOS</p> <p>1390 120</p>	<p>MES: JUNIO</p> <p>TOTAL COMPRAS DESCUENTOS</p>
<p>MES: AGOSTO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: JULIO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: JULIO</p> <p>TOTAL COMPRAS DESCUENTOS</p>
<p>MES: SEPTIEMBRE</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: AGOSTO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: AGOSTO</p> <p>TOTAL COMPRAS DESCUENTOS</p>
<p>MES: AGOSTO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: AGOSTO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	
<p>MES: SEPTIEMBRE</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>MES: SEPTIEMBRE</p> <p>TOTAL COMPRAS DESCUENTOS</p>	
<p>-----</p> <p>Cliente: JKM S.R.L</p> <p>MES: FEBRERO</p> <p>TOTAL COMPRAS DESCUENTOS</p>	<p>-----</p>	
<p>MES: MARZO</p> <p>TOTAL COMPRAS DESCUENTOS</p>		
<p>MES: ABRIL</p> <p>TOTAL COMPRAS DESCUENTOS</p>		
<p>MES: MAYO</p> <p>TOTAL COMPRAS DESCUENTOS</p> <p>300 0</p>		
<p>MES: JUNIO</p> <p>TOTAL COMPRAS DESCUENTOS</p>		
<p>MES: JULIO</p> <p>TOTAL COMPRAS DESCUENTOS</p>		