**TRABAJO PRÁCTICO**

**MODELO DE PUNTO DE VENTA**

**AÑO 2021**

**Integrantes del grupo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cédula** | **Apellido** | **Nombre** | **Sección** |
| **6129422** | **Amarilla Leguizamón** | **Óscar Alexander** | **NA** |
| **4659580** | **Gómez Cardenas** | **Eduardo David** | **NA** |
| **4662525** | **Medina Recalde** | **Oliver Isaías** | **NA** |

**PARTE 2**

**DDL, VISTAS, SEGURIDAD**

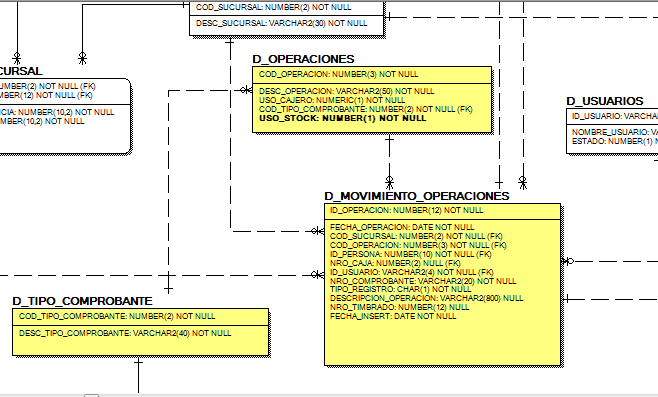
1. Cree el espacio de tabla BASECONTABLE con un tamaño inicial de 100MB y que se autoextienda.
2. Cree en la base de datos las tablas nuevas mostradas en el DER\_TP2, asignándoles el tablespace BASECONTABLE. Considere las restricciones de integridad y los comentarios que se muestran en el siguiente cuadro:

| **Nombre** | **¿Nulo?** | **Tipo** | **COMENTANTARIOS /RESTRICCIONES** |
| --- | --- | --- | --- |
| **D\_ASIENTO\_CABECERA** | **Cabecera de asientos contables** | | |
| ID\_ASIENTO | NOT NULL | NUMBER(10) | Identificación del asiento (Identity obligatorio) (PK) |
| FECHA\_ASIENTO | NOT NULL | DATE | Fecha del asiento |
| CONCEPTO | NOT NULL | VARCHAR2(500) | Concepto del asiento |
| **D\_ASIENTO\_DETALLE** | **Lineas de detalle del asiento contable** | | |
| ID\_ASIENTO | NOT NULL | NUMBER(10) | Identificación del asiento (Identity obligatorio) (PK) |
| SECUENCIA | NOT NULL | NUMBER(4) | Secuencia del asiento (PK) |
| DEBE\_HABER |  | VARCHAR2(20) | Indica si el importe del asiento afecta en el DEBE ('D') o en el HABER ('H'). |
| IMPORTE | NOT NULL | NUMBER(9) | Importe del asiento |
| CODIGO\_CTA | NOT NULL | NUMBER(7) | Código de la cuenta |
| **D\_CUENTAS\_CONTABLES** | **Plan de Cuentas Contable. Esta tabla debe crearla a partir de la tabla B\_CUENTAS de nuestro modelo empresarial, y luego aplicar los constraints** | | |
| CODIGO\_CTA | NOT NULL | NUMBER(7) | código de la cuenta (PK) |
| NOMBRE\_CTA |  | VARCHAR2(40) | Nombre de la cuenta |
| ID\_TIPO | NOT NULL | NUMBER(4) | Determina tipo: 1(Activo) 2(Pasivo) 3(Patrimonio) 4 (Gastos) 5 (Ingresos) |
| NIVEL | NOT NULL | NUMBER(1) | Nivel de la cuenta en la jerarquía: nivel 1: primer dígito, nivel 2: Segundo dígito, nivel 3 (tercer digito) nivel 4 (cuarto y quinto digito) nivel 5 (sexto y séptimo dígito) |
| ORDEN |  | VARCHAR2(1) | D: Deudor y C:Acreedor |
| IMPUTABLE | NOT NULL | VARCHAR2(1) | Indica si una cuenta es imputable ('S') o no ('N'). Cuando es imputable, una cuenta acepta asientos contables. |
| FECHA\_APERTURA |  | DATE | Fecha de apertura de la cuenta |
| CODIGO\_EQUIVALENTE |  | NUMBER(8) | Código de producto, de cuenta, de cliente según el tipo de cuenta |
| **D\_PLANTILLA\_ASIENTO** | **Template de Asientos contables** | | |
| ID\_PLANTILLA | NOT NULL | NUMBER(10) | Identificación de la planilla. Identity por default (PK) |
| NOMBRE | NOT NULL | VARCHAR2(60) | Nombre de la plantilla |
| ACTIVA | NOT NULL | VARCHAR2(1) | Indica si la plantilla está activa ('S') o no ('N') |
| **D\_PLANTILLA\_DETALLE** | **Lineas de detalle del Template de Asientos contables** | | |
| ID\_PLANTILLA | NOT NULL | NUMBER(10) | Identificación. PK |
| SECUENCIA | NOT NULL | NUMBER(4) | Secuencia del asiento PK |
| DEBE\_HABER |  | VARCHAR2(20) | Indica si el importe del asiento afecta en el DEBE ('D') o en el HABER ('H'). |
| IMPORTE | NOT NULL | NUMBER(9) | Importe del asiento |
| CODIGO\_CTA |  | NUMBER(7) | código de la cuenta |
| CATEGORIA | NOT NULL | VARCHAR2(20) | Categorización en la que se aplicará la dinámica contable: Ej: APERTURA, CIERRE, VENTA, COMPRA, etc |

SOLUCIÓN

|  |
| --- |
| -- Enunciado 1  CREATE TABLESPACE BASECONTABLE  DATAFILE 'C:\oradata\TP2.dbf' SIZE 100M  DEFAULT STORAGE (INITIAL 100M NEXT 50K)  ONLINE;  ---------------  -- Enunciado 2  -- Creacion de tablas  CREATE TABLE D\_ASIENTO\_CABECERA (  ID\_ASIENTO NUMBER(10) GENERATED ALWAYS AS IDENTITY,  FECHA\_ASIENTO DATE NOT NULL,  CONCEPTO VARCHAR(500) NOT NULL,  CONSTRAINT D\_ASIENTO\_CABECERA\_pk PRIMARY KEY (ID\_ASIENTO)  );  CREATE TABLE D\_PLANTILLA\_ASIENTO (  ID\_PLANTILLA NUMBER(10) GENERATED ALWAYS AS IDENTITY,  NOMBRE VARCHAR2(60) NOT NULL,  ACTIVA VARCHAR2(1) NOT NULL CONSTRAINT ACT\_CONSTRAINT  CHECK (  ACTIVA IN ('S', 'N')  ),  CONSTRAINT D\_PLANTILLA\_ASIENTO\_PK PRIMARY KEY (ID\_PLANTILLA)  );  CREATE TABLE D\_CUENTAS\_CONTABLES (  CODIGO\_CTA NUMBER(7) NOT NULL,  CODIGO\_EQUIVALENTE NUMBER(8),  ID\_TIPO NUMBER(4) NOT NULL CONSTRAINT IT\_CONSTRAINT  CHECK (  ID\_TIPO IN (1,2,3,4,5)  ),  NIVEL NUMBER(1) NOT NULL CONSTRAINT NVL\_CONSTRAINT  CHECK (  NIVEL IN (1,2,3,4,5)  ),  ORDEN VARCHAR2(1) CONSTRAINT ORD\_CONSTRAINT  CHECK (  ORDEN IN ('D', 'C')  ),  NOMBRE\_CTA VARCHAR2(40),  IMPUTABLE VARCHAR2(1) NOT NULL CONSTRAINT IMP\_CONSTRAINT  CHECK (  IMPUTABLE IN ('S', 'N')  ),  FECHA\_APERTURA DATE,  CONSTRAINT D\_CUENTAS\_CONTABLES\_PK PRIMARY KEY (CODIGO\_CTA)  );  CREATE TABLE D\_PLANTILLA\_DETALLE (  SECUENCIA NUMBER(4) NOT NULL,  ID\_PLANTILLA NUMBER(10) NOT NULL,  CODIGO\_CTA NUMBER(7) NOT NULL,  CATEGORIA VARCHAR2(20) NOT NULL,  DEBE\_HABER VARCHAR2(20) CONSTRAINT DH\_CONSTRAINT  CHECK (  DEBE\_HABER IN ('D', 'H')  ),  IMPORTE NUMBER(9) NOT NULL,  CONSTRAINT D\_PLANTILLA\_DETALLE\_PK PRIMARY KEY (SECUENCIA, ID\_PLANTILLA)  );  CREATE TABLE D\_ASIENTO\_CABECERA (  ID\_ASIENTO NUMBER(10) NOT NULL,  FECHA\_ASIENTO DATE NOT NULL,  CONCEPTO VARCHAR2(500) NOT NULL,  CONSTRAINT D\_ASIENTO\_CABECERA\_PK PRIMARY KEY (ID\_ASIENTO)  );  CREATE TABLE D\_ASIENTO\_DETALLE (  ID\_ASIENTO NUMBER(10) GENERATED ALWAYS AS IDENTITY,  SECUENCIA NUMBER(4) NOT NULL,  DEBE\_HABER VARCHAR2(20) CONSTRAINT DHD\_CONSTRAINT  CHECK (  DEBE\_HABER IN ('D', 'H')  ),  IMPORTE NUMBER(9) NOT NULL,  CODIGO\_CTA NUMBER(7) NOT NULL,  CONSTRAINT D\_ASIENTO\_DETALLE\_PK  PRIMARY KEY (ID\_ASIENTO, SECUENCIA)  );  ALTER TABLE D\_PLANTILLA\_DETALLE ADD CONSTRAINT D\_PLANTILLA\_ASIENTO\_D\_PLANT799  FOREIGN KEY (ID\_PLANTILLA)  REFERENCES D\_PLANTILLA\_ASIENTO (ID\_PLANTILLA)  NOT DEFERRABLE;  ALTER TABLE D\_PLANTILLA\_DETALLE ADD CONSTRAINT D\_CUENTAS\_CONTABLES\_D\_PLANT523  FOREIGN KEY (CODIGO\_CTA)  REFERENCES D\_CUENTAS\_CONTABLES (CODIGO\_CTA)  NOT DEFERRABLE;  ALTER TABLE D\_ASIENTO\_DETALLE ADD CONSTRAINT D\_CUENTAS\_CONTABLES\_D\_ASIEN635  FOREIGN KEY (CODIGO\_CTA)  REFERENCES D\_CUENTAS\_CONTABLES (CODIGO\_CTA)  NOT DEFERRABLE;  ALTER TABLE D\_ASIENTO\_DETALLE ADD CONSTRAINT D\_ASIENTO\_CABECERA\_D\_ASIENT658  FOREIGN KEY (ID\_ASIENTO)  REFERENCES D\_ASIENTO\_CABECERA (ID\_ASIENTO)  NOT DEFERRABLE;  alter table D\_ASIENTO\_CABECERA move tablespace BASECONTABLE;  alter table D\_CUENTAS\_CONTABLES move tablespace BASECONTABLE;  alter table D\_PLANTILLA\_ASIENTO move tablespace BASECONTABLE;  alter table D\_PLANTILLA\_DETALLE move tablespace BASECONTABLE;  alter table D\_ASIENTO\_DETALLE move tablespace BASECONTABLE; |

* 1. Altere la tabla D\_OPERACIONES para relacionarla con el tipo de comprobante. De esa manera, el tipo de comprobante utilizado ya estará determinado por la operación. El modelo deberá lucir de la siguiente manera:



Tenga en cuenta que el atributo cod\_tipo\_comprobante se suprime de la tabla **D\_MOVIMIENTO\_OPERACIONES**, y se agrega a la tabla **D\_OPERACIONES**, estableciendo la relación con la clave primaria de la tabla D\_TIPO\_COMPROBANTE. Para agregar dicha columna nueva a la tabla de D\_OPERACIONES puede indicar por defecto el cod\_tipo\_comprobante = 1 (FACTURA).

Se agrega también el campo **USO\_STOCK** que serán 1: Resta y 2: Suma.

SOLUCIÓN

|  |
| --- |
| --Se elimina el constraint  ALTER TABLE D\_MOVIMIENTO\_OPERACIONES DROP COLUMN COD\_TIPO\_COMPROBANTE CASCADE CONSTRAINTS;  -- Se agrega la fk a d\_operaciones  ALTER TABLE D\_OPERACIONES  ADD COD\_TIPO\_COMPROBANTE NUMBER(2) DEFAULT 1 NOT NULL;  ALTER TABLE D\_OPERACIONES  ADD CONSTRAINT FK\_TIPO\_COMBROBANTE  foreign key (COD\_TIPO\_COMPROBANTE)  references D\_TIPO\_COMPROBANTE(COD\_TIPO\_COMPROBANTE)  on delete cascade;  -- Se visualiza los constrain  SELECT CONSTRAINT\_NAME, TABLE\_NAME, R\_CONSTRAINT\_NAME FROM USER\_CONSTRAINTS where table\_name='D\_OPERACIONES';  -- Se agrega el campo USO\_STOCK  ALTER TABLE D\_OPERACIONES  ADD USO\_STOCK NUMBER(1) DEFAULT 1 CONSTRAINT CHK\_STOCK  CHECK (  USO\_STOCK IN (1,2)  ) NOT NULL; |

1. Una vista materializada que permita obtener el informe de cierre diario de las cajas (monto pagado). La vista debe lucir así:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **CAJA NRO 1** | **CAJA NRO 2** | **CAJA NRO 3** | **CAJA NRO 4** |
| EFECTIVO |  |  |  |  |
| EFECTIVO PROMO |  |  |  |  |
| TARJETA ITAU |  |  |  |  |
| TARJETA DE LA CASA |  |  |  |  |
| TARJETA CREDITO |  |  |  |  |
| **TOTAL** |  |  |  |  |

Debe iniciar el refrescado a las **20:00** del día de hoy y actualizarse **cada día** en el mismo horario.

SOLUCIÓN

|  |
| --- |
| CREATE MATERIALIZED VIEW cierre\_diario\_cajas  TABLESPACE BASECONTABLE  BUILD IMMEDIATE  REFRESH START WITH ROUND(SYSDATE-1)+20/24 NEXT ROUND(SYSDATE+1)+20/24  AS  SELECT  FP.DESC\_FORMA\_PAGO AS FORMA\_PAGO,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 1  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA1,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 2  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA2,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 3  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA3,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 4  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA4  FROM D\_FORMA\_PAGO FP  LEFT JOIN D\_PAGO\_OPERACION PO ON FP.COD\_FORMA\_PAGO = PO.COD\_FORMA\_PAGO  LEFT JOIN D\_MOVIMIENTO\_OPERACIONES MVO ON MVO.ID\_OPERACION = PO.ID\_OPERACION  LEFT JOIN D\_CAJAS CJ ON CJ.NRO\_CAJA = MVO.NRO\_CAJA  GROUP BY FP.DESC\_FORMA\_PAGO  UNION ALL  SELECT 'TOTAL' AS FORMA\_PAGO,  SUM(CAJA1) AS CAJA1,  SUM(CAJA2) AS CAJA2,  SUM(CAJA3) AS CAJA3,  SUM(CAJA4) AS CAJA4  FROM (  SELECT  FP.DESC\_FORMA\_PAGO AS FORMA\_PAGO,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 1  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA1,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 2  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA2,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 3  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA3,  SUM(  CASE  WHEN CJ.NRO\_CAJA = 4  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS CAJA4,  SUM(  CASE  WHEN CJ.NRO\_CAJA IN (1,2,3,4)  THEN PO.IMPORTE\_PAGO  ELSE 0  END  ) AS TOTAL  FROM D\_FORMA\_PAGO FP  LEFT JOIN D\_PAGO\_OPERACION PO ON FP.COD\_FORMA\_PAGO = PO.COD\_FORMA\_PAGO  LEFT JOIN D\_MOVIMIENTO\_OPERACIONES MVO ON MVO.ID\_OPERACION = PO.ID\_OPERACION  LEFT JOIN D\_CAJAS CJ ON CJ.NRO\_CAJA = MVO.NRO\_CAJA  GROUP BY FP.DESC\_FORMA\_PAGO  ORDER BY FP.DESC\_FORMA\_PAGO  ); |

1. Seguridad
   1. Habilite el SPOOL
   2. Genere un SCRIPT con una sentencia SELECT, que cree sinónimos para todas las tablas nuevas creadas del modelo (Que inicien con D\_). El sinónimo será el mismo nombre de la tabla. Posteriormente ejecute el resultado como script en la BD.
   3. Cree el rol R\_CONS
   4. Al igual que en el punto b), genere otro SCRIPT para conceder el permiso de SELECT al rol creado sobre todas las tablas del modelo.

SOLUCIÓN

|  |
| --- |
| -- ENUNCIADO 4  -- Item A  SPOOL temp.sql;  --Item B  SELECT 'CREATE SYNONYM "' || SUBSTR(a.table\_name, 3) || '" FOR "' || a.owner || '"."' || a.table\_name || '";'  FROM dba\_tables a  WHERE TABLESPACE\_NAME ='BASECONTABLE'  AND a.table\_name LIKE 'D\_%';  SPOOL OFF;  @temp.sql  --- Item C  ALTER SESSION SET "\_ORACLE\_SCRIPT"=true;  CREATE ROLE R\_CONS;  ALTER SESSION SET "\_ORACLE\_SCRIPT"=false;  --- Item D  SPOOL temp1.sql;  SELECT 'GRANT SELECT ON ' || a.table\_name || ' TO R\_CONS;'  FROM dba\_tables a  WHERE TABLESPACE\_NAME ='BASECONTABLE'  AND a.table\_name LIKE 'D\_%';  SPOOL OFF;  @temp1.sql  --- Comprobación de que se han creado los roles  SELECT TABLE\_NAME, privilege FROM ROLE\_TAB\_PRIVS WHERE ROLE = 'R\_CONS'; |

**PUNTUACIÓN**

|  |  |  |
| --- | --- | --- |
| **Ejercicio** | **Criterio** | **Puntos** |
| 1 | Tablespace | 2 |
| 2 | Las tablas deben estar completas con los constraints (PK, FK, Checks) y comentarios |  |
| 2 | D\_CUENTAS\_CONTABLES | 3 |
|  | D\_ASIENTO\_CABECERA | 1,5 |
|  | D\_ASIENTO\_DETALLE | 1,5 |
|  | D\_PLANTILLA\_ASIENTO | 1 |
|  | D\_PLANTILLA\_DETALLE | 1 |
| 3 | Vista materializada | 5 |
| 4 | a) spool |  |
|  | b) sinónimo | 2 |
|  | c) rol | 1 |
|  | d) granteo | 2 |
|  | **TOTAL** | **20** |