INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS QUERÉTARO

Lab_BDA_ETL-3

DBA

Daniel Cu Sánchez - A01703613

ITESM Campus Querétaro Bases de Datos Avanzadas Lab: Ejemplos de ETL Fecha de entrega: martes 26 abril de 2022

*Para obtener la calificación del presente laboratorio, es necesario tener la evidencia del desarrollo del mismo. La evidencia será un archivo PDF el cual contendrá las imágenes de pantallas que se van a ir indicando en el desarrollo del laboratorio como ***EVIDENCIA***. El archivo lo suben a canvas en la tarea nombrada "***Lab: Éjemplos ETL" El nombre del archivo será: Matricula_LabETL.pdf*.

En el presente laboratorio revisaremos varios ejemplos que puedan ayudar con el proceso **ETL** de sus **cubos** del **DWH**.

Parte I. Configurar el ambiente

Para esta parte sigue las instrucciones del profesor

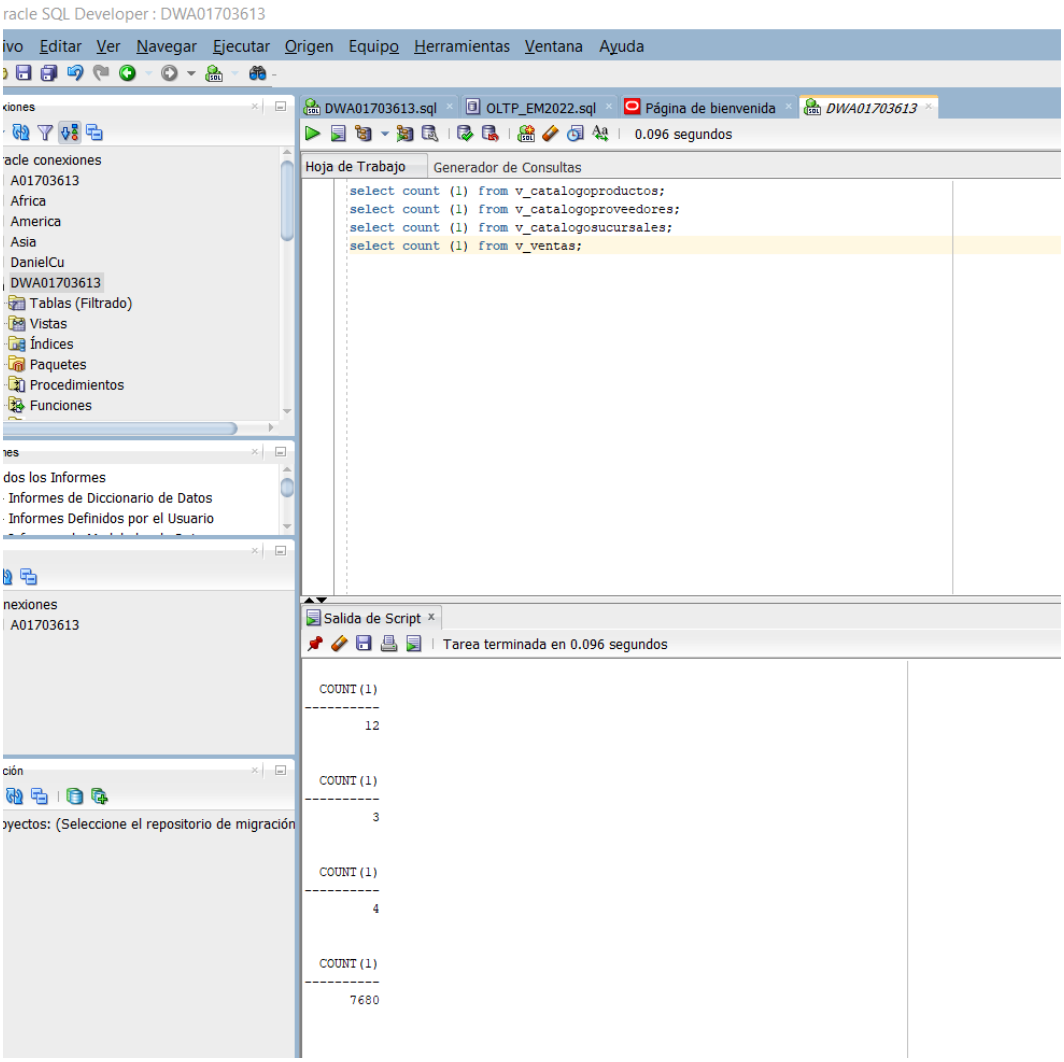
a) Para facilitar el manejo del DWH ejecuta el script **OLTP_EM2022.sql** en el esquema de tu matricula. El script creara los objetos (base de datos transaccional). Una vez creados los objetos carga los datos a la tabla VENTAS a partir del archivo **datos_ventas.csv**
b) Adicional en la base de datos OLAP (base de datos del repositorio del DWH con la cuenta DWHA0XXXXXX) ejecutar el script **OLAP_EM2022.sql** el cual creara los objetos que nos ayudaran en el desarrollo del presente laboratorio.

Parte II. Revisión de vistas, tablas y procedimientos.

Conéctate al esquema **DWHMatricula -OLAP-** y ejecuta las siguientes instrucciones

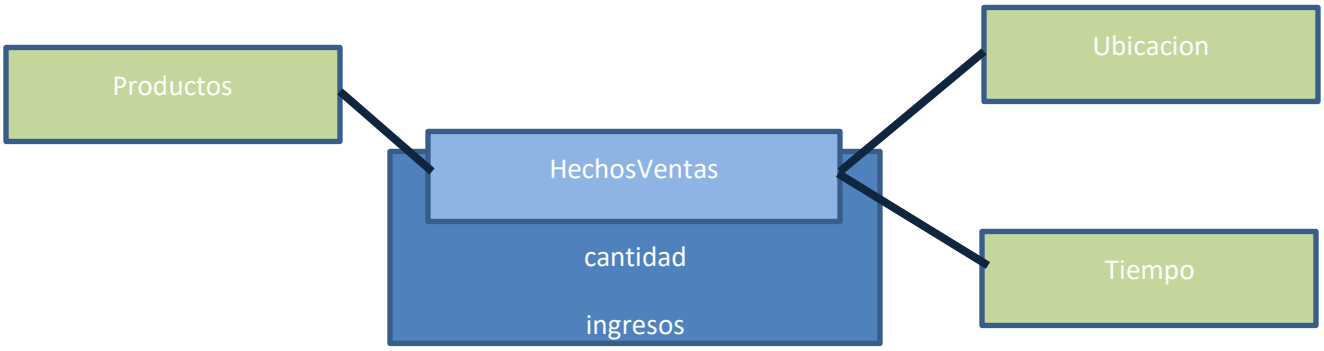
```
select count (1) from v_catalogoproductos;
select count (1) from v_catalogoproveedores;
select count (1) from v_catalogosucursales;
select count (1) from v_ventas;
```

*****EVIDENCIA*****



Revisa la estructura de las vistas.

Considerando el siguiente **esquema estrella** para un **cubo** de un **datawarehouse**:



Para la **dimensión** de “**Productos**”, vamos a tener los siguientes atributos:

- *Descripción del producto*
- *Nombre del proveedor*
- *Código del producto (base de datos transaccional)*

Para la **dimensión** “**Ubicación**”, vamos a tener los siguientes atributos:

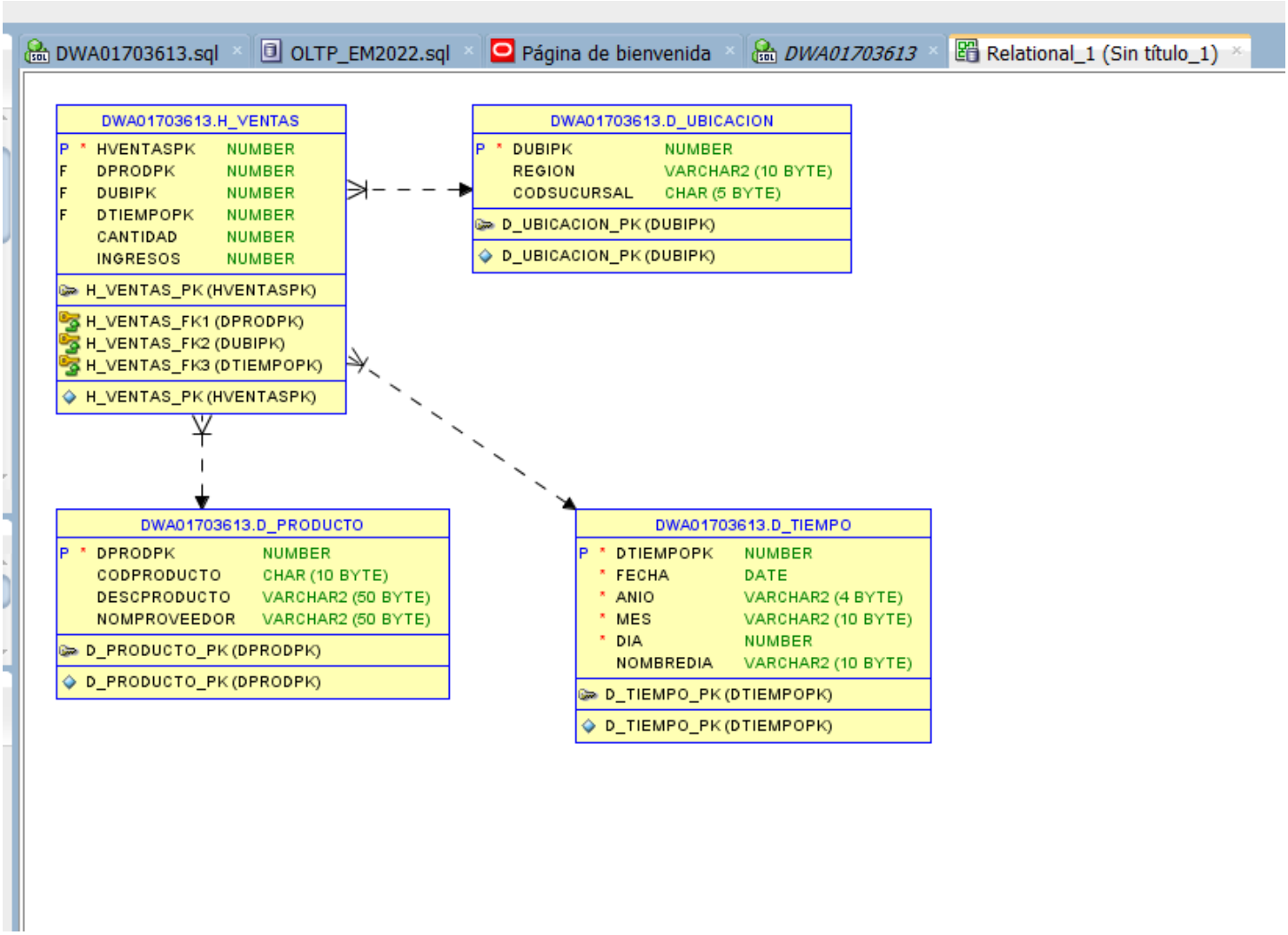
- *Region*
- *Código de sucursal (base de datos transaccional)*

Para la **dimensión** “**Tiempo**”, vamos a tener los siguientes atributos:

- Fecha
- Año
- Mes
- Día
- Nombre del día

Revisa las estructuras de las tablas **D_PRODUCTOS, D_TIEMPO, D_UBICACION** y **H_VENTAS.** De la misma manera revisa los **constraints** entre cada una de las tablas.

*****EVIDENCIA*****
Captura la pantalla con la estructura de las tablas y las relaciones entre las dimensiones y hechos



Parte III. Implementación de procedimiento de ETL para tablas de dimensiones.

Para mantener **actualizadas** las **tablas** de **dimensiones** es necesario extraer la información de las bases de datos transaccionales. En nuestro caso para facilitar la extracción creamos las vistas que nos ayudaran en el proceso. Solamente vamos a insertar los registros "**nuevos**" de nuestra base de datos transaccional. Para esta práctica consideramos que no debemos borrar ningún registro del **DWH** aun cuando se borren en la base de datos transaccional, esto para mantener **información** "**histórica**" en el **DWH**.

Para la **dimensión producto** actualiza el procedimiento **ACTUALIZA_PRODUCTO** de la siguiente manera (recuerda estar ya conectado al esquema dwhMatricula):

```
create or replace PROCEDURE ACTUALIZA_PRODUCTO AS
BEGIN
  insert into d_producto
  select seq_d_producto.nextval, codproducto,descripcion, razonsocial
  from v_catalogoproductos cp, v_catalogoproveedores cv
  where cp.rfcproveedor = cv.rfcproveedor
  and cp.codproducto not in (select codproducto from d_producto);
  commit;
END ACTUALIZA_PRODUCTO;
```

Para la **dimensión ubicación** actualiza el procedimiento **ACTUALIZA_UBICACION** a partir de la siguiente consulta:

```
select seq_d_ubicacion.nextval,region,codsucursal
from v_catalogosucursales cs
where cs.codsucursal not in (select codsucursal from d_ubicacion);
```

Ejecuta los **dos procedimientos** que acabas de modificar y en este punto ya tenemos pobladas las dos tablas de dimensiones del cubo.

*****EVIDENCIA*** REALIZA LA CONSULTA A CADA UNA DE LAS TABLAS QUE SE ACABAN DE POBLAR Y TOMA LA CAPTURA DE LA MISMA – CONSULTA Y DATOS Y/O PESTAÑA DE DATOS DESDE SQL DEVELOPER-**

DWA01703613.sql | Página de bienvenida | DWA01703613

Hoja de Trabajo | Generador de Consultas

```
create or replace PROCEDURE ACTUALIZA_PRODUCTO AS
BEGIN
  insert into d_producto
  select seq_d_producto.nextval, codproducto, descripcion, razonsocial
  from v_catalogoproductos cp, v_catalogoproveedores cv
  where cp.rfcproveedor = cv.rfcproveedor
  and cp.codproducto not in (select codproducto from d_producto);
commit;
END ACTUALIZA_PRODUCTO;


SELECT * FROM dwa01703613.d_producto;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 12 en 0.007 segundos

	DPRODPK	CODPRODUCTO	DESCPRODUCTO	NOMPROVEEDOR
1	1	99900003	Pay de nuez	Bimbo-Marinela S.A. de C.V.
2	2	99900004	Canelitas	Bimbo-Marinela S.A. de C.V.
3	3	77700003	Victoria	Embotelladora Victoria del Centro
4	4	88800001	Sabritas	Pepsi Co. De Mexico
5	5	99900001	Submarinos	Bimbo-Marinela S.A. de C.V.
6	6	99900002	Pinguinos	Bimbo-Marinela S.A. de C.V.
7	7	99900006	Gansito	Bimbo-Marinela S.A. de C.V.
8	8	77700004	Sprite	Embotelladora Victoria del Centro
9	9	88800003	Sabritones	Pepsi Co. De Mexico
10	10	77700002	Manzana Lift	Embotelladora Victoria del Centro
11	11	88800002	Fritos	Pepsi Co. De Mexico
12	12	77700001	Coca cola	Embotelladora Victoria del Centro

DWA01703613.sql | Página de bienvenida | *DWA01703613*

Hoja de Trabajo | Generador de Consultas

```
create or replace PROCEDURE ACTUALIZA_UBICACION AS
BEGIN
  insert into dwa01703613.d_ubicacion (dubipk, region, codsucursal)
  select dwa01703613.seq_d_ubicacion.nextval,region, codsucursal
  from dwa01703613.v_catalogosucursales cs
  where cs.codsucursal not in (select codsucursal from dwa01703613.d_ubicacion);
commit;
END ACTUALIZA_UBICACION;

SELECT * FROM dwa01703613.d_ubicacion;
```

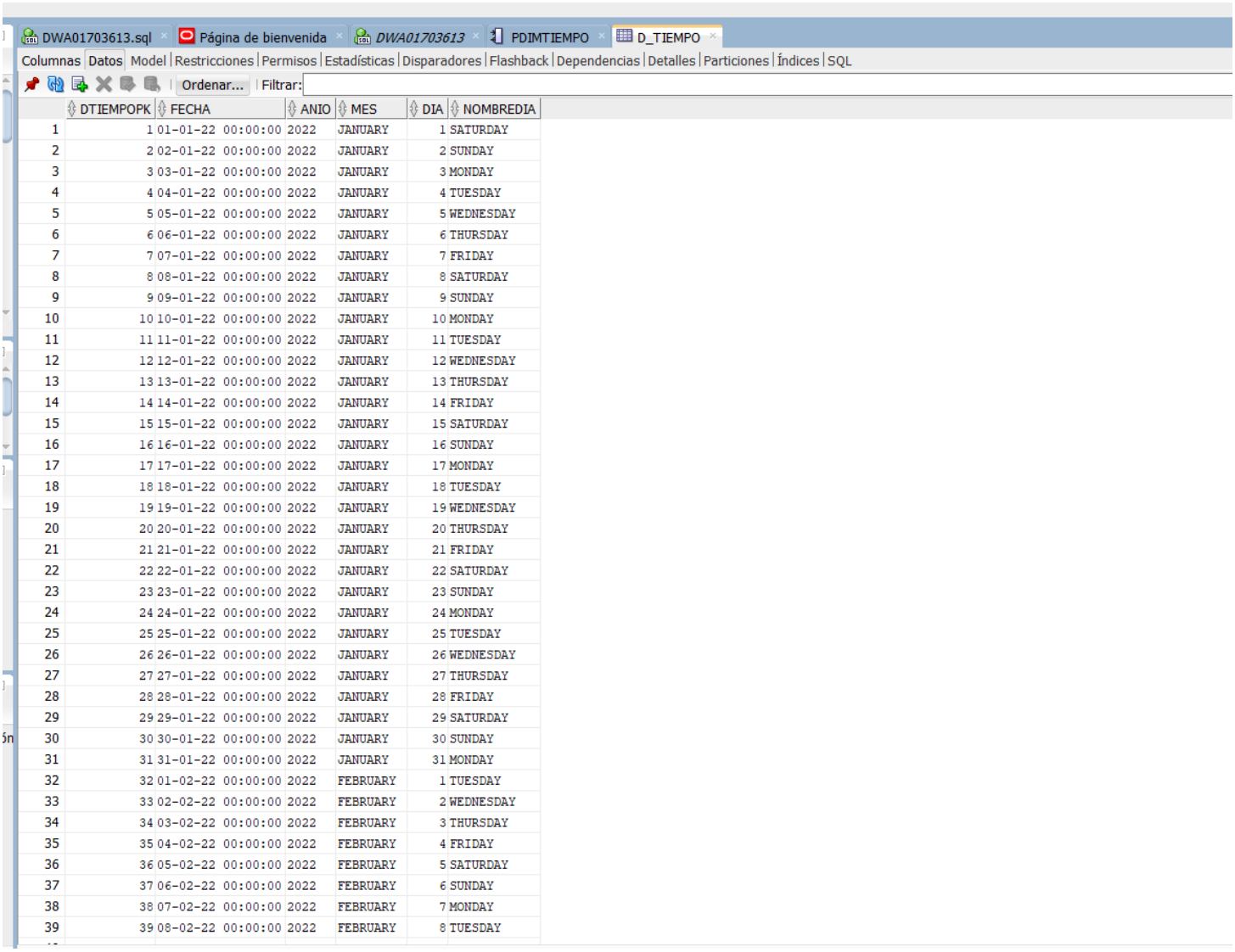
Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 4 en 0.009 segundos

	DUBIPK	REGION	CODSUCURSAL
1	1	Bajio	QRO
2	2	Centro	MEX
3	3	Norte	MTY
4	4	Occidente	GDA

Para la **dimensión tiempo** es necesario ejecutar el procedimiento **PDIMTIEMPO** para poblar con datos entre las fechas **1 de enero de 2022** y **21 de abril de 2022**.

*****EVIDENCIA*** REALIZA LA CONSULTA A LA TABLA S QUE SE ACABAN DE POBLAR Y TOMA LA CAPTURA DE LA MISMA – CONSULTA Y DATOS Y/O PESTAÑA DE DATOS DESDE SQL DEVELOPER-**



	DTIEMPOPK	FECHA	ANIO	MES	DIA	NOMBREDIA
1	1	01-01-22 00:00:00	2022	JANUARY	1	SATURDAY
2	2	02-01-22 00:00:00	2022	JANUARY	2	SUNDAY
3	3	03-01-22 00:00:00	2022	JANUARY	3	MONDAY
4	4	04-01-22 00:00:00	2022	JANUARY	4	TUESDAY
5	5	05-01-22 00:00:00	2022	JANUARY	5	WEDNESDAY
6	6	06-01-22 00:00:00	2022	JANUARY	6	THURSDAY
7	7	07-01-22 00:00:00	2022	JANUARY	7	FRIDAY
8	8	08-01-22 00:00:00	2022	JANUARY	8	SATURDAY
9	9	09-01-22 00:00:00	2022	JANUARY	9	SUNDAY
10	10	10-01-22 00:00:00	2022	JANUARY	10	MONDAY
11	11	11-01-22 00:00:00	2022	JANUARY	11	TUESDAY
12	12	12-01-22 00:00:00	2022	JANUARY	12	WEDNESDAY
13	13	13-01-22 00:00:00	2022	JANUARY	13	THURSDAY
14	14	14-01-22 00:00:00	2022	JANUARY	14	FRIDAY
15	15	15-01-22 00:00:00	2022	JANUARY	15	SATURDAY
16	16	16-01-22 00:00:00	2022	JANUARY	16	SUNDAY
17	17	17-01-22 00:00:00	2022	JANUARY	17	MONDAY
18	18	18-01-22 00:00:00	2022	JANUARY	18	TUESDAY
19	19	19-01-22 00:00:00	2022	JANUARY	19	WEDNESDAY
20	20	20-01-22 00:00:00	2022	JANUARY	20	THURSDAY
21	21	21-01-22 00:00:00	2022	JANUARY	21	FRIDAY
22	22	22-01-22 00:00:00	2022	JANUARY	22	SATURDAY
23	23	23-01-22 00:00:00	2022	JANUARY	23	SUNDAY
24	24	24-01-22 00:00:00	2022	JANUARY	24	MONDAY
25	25	25-01-22 00:00:00	2022	JANUARY	25	TUESDAY
26	26	26-01-22 00:00:00	2022	JANUARY	26	WEDNESDAY
27	27	27-01-22 00:00:00	2022	JANUARY	27	THURSDAY
28	28	28-01-22 00:00:00	2022	JANUARY	28	FRIDAY
29	29	29-01-22 00:00:00	2022	JANUARY	29	SATURDAY
30	30	30-01-22 00:00:00	2022	JANUARY	30	SUNDAY
31	31	31-01-22 00:00:00	2022	JANUARY	31	MONDAY
32	32	01-02-22 00:00:00	2022	FEBRUARY	1	TUESDAY
33	33	02-02-22 00:00:00	2022	FEBRUARY	2	WEDNESDAY
34	34	03-02-22 00:00:00	2022	FEBRUARY	3	THURSDAY
35	35	04-02-22 00:00:00	2022	FEBRUARY	4	FRIDAY
36	36	05-02-22 00:00:00	2022	FEBRUARY	5	SATURDAY
37	37	06-02-22 00:00:00	2022	FEBRUARY	6	SUNDAY
38	38	07-02-22 00:00:00	2022	FEBRUARY	7	MONDAY
39	39	08-02-22 00:00:00	2022	FEBRUARY	8	TUESDAY

Parte IV Implementación de procedimiento de ETL para la tabla de hechos.

Para la implementación de la **tabla** de **hechos** vamos a tomar las siguientes consideraciones:

- Debemos especificar la **fecha** de **inicio** y la **fecha** de **fin** de la extracción de datos. Esto puede ser muy útil para una **carga inicial** de datos y después para **cargas incrementales** (reducción de cargas de trabajo tanto en las bases de datos transaccionales como la del DWH), por ejemplo, cargas diarias o semanales.
- Para evitar la carga de **datos** "**duplicados**" o **errores** en los **constraints** para periodos de tiempo ya capturados, los procedimientos de cargas de hechos deben poder actualizar y/o sustituir la información ya existente. Una manera fácil de realizar esto es **borrando** antes de la inserción los registros que correspondan al periodo de tiempo que se va a actualizar. En nuestro caso el procedimiento de carga debe **iniciar con** el **borrado** de los **registros** que corresponden al periodo del tiempo a insertar.
- Los indicadores de cantidad e ingresos se calculan de la siguiente manera ○ **cantidad** ❼ sum(ventas.cantidad)
    - **ingresos** ❼ sum(ventas.cantidad*ventas.precio)
- Todas las **dimensiones** deben **agruparse** juntas, considerando el tiempo, producto y ubicación.
- Para el caso de la dimensión tiempo se debe usar la **función trunc** de la fecha para que sea fácil la comparación de las fechas.

Teniendo en cuenta las consideraciones anteriores, **escribe** una **consulta** que obtenga el **identificador** del **producto**, **ubicación** y **tiempo** de las **dimensiones** y se calcule los **indicadores** a partir de la **vista** de **ventas**. Usa la **función** de **agrupación** para que puedas realizar la consulta. Las condiciones deben cumplir con lo siguiente: ○ Para la dimensión producto, usar el correspondiente código de producto transaccional con su correspondiente registro en la vista de ventas. ○ Para la dimensión ubicación, usar el correspondiente código de ubicación transaccional con su correspondiente registro en la vista de ventas.

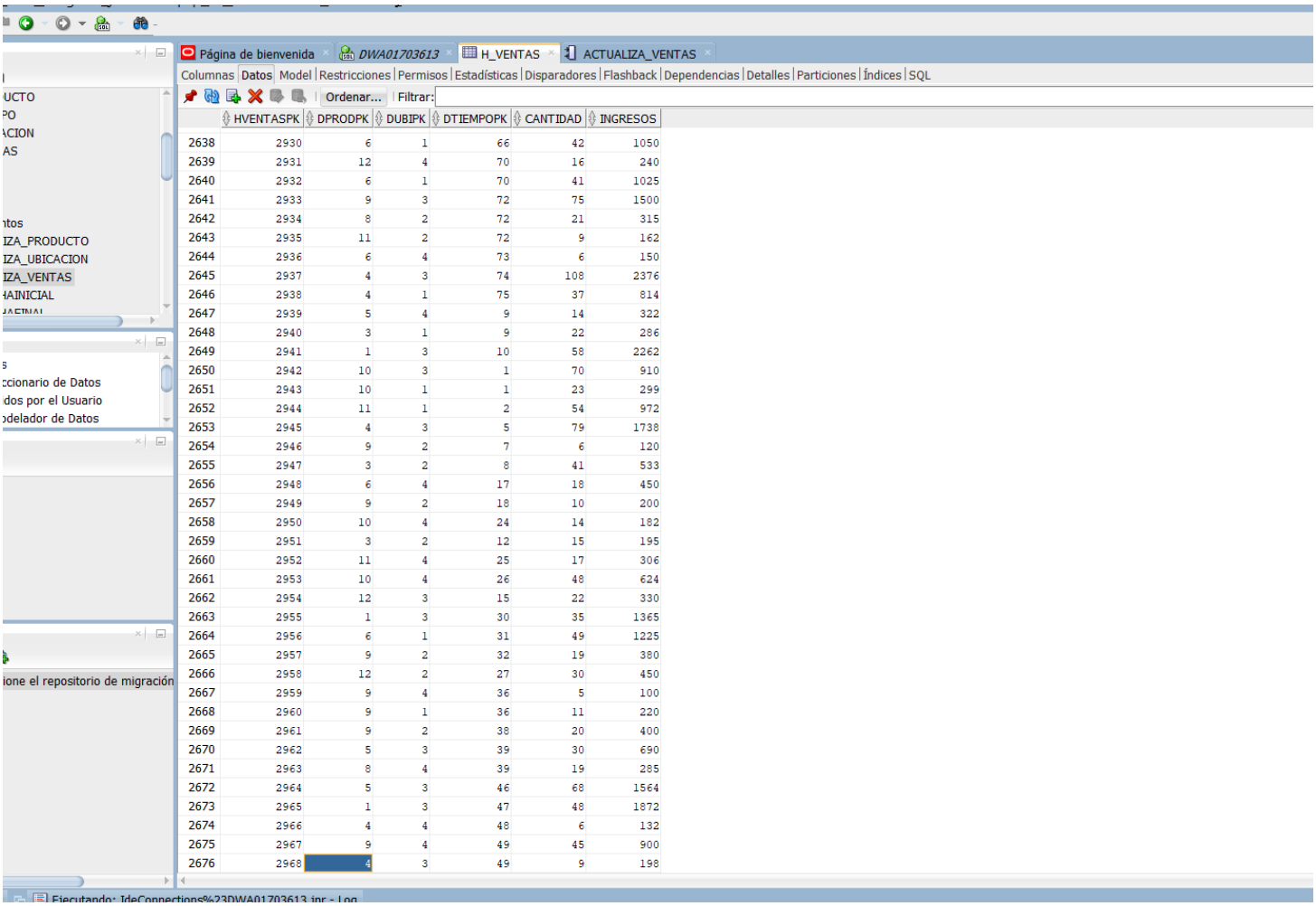
- Para la dimensión tiempo usa la función **between fechainicio and fechafinal** para obtener los registros de la vista de ventas y realizar el **join** con la función **trunc** a la correspondiente columna de la dimensión del tiempo.

Una vez que tengas la consulta, ajusta el procedimiento **ACTUALIZA_VENTAS** con la siguiente estructura:

```
create or replace
PROCEDURE ACTUALIZA_VENTAS
(
  FECHAINICIAL IN DATE
, FECHAFINAL IN DATE
) AS
  vFechaInicial date;
  vFechaFinal   date;
  vdProPk       number;
  vdTiempoPk    number;
  vdUbiPk       number;
  v_cantidad    number;
  v_ingresos    number;

  cursor c_tiempo is
  select dtiempopk
  from d_tiempo
  where fecha between vFechaInicial and vFechaFinal;

  cursor c_ventas is
  -- código de la consulta desarrollada_


BEGIN
  vFechaInicial := FECHAINICIAL;
  vFechaFinal   := FECHAFINAL;

  open c_tiempo;
  LOOP
    fetch c_tiempo into vdTiempoPk;
    exit when c_tiempo%NOTFOUND;
    delete from h_ventas where dtiempopk=vdTiempoPk;
    commit;
  END LOOP;
  close c_tiempo;

  open c_ventas;
  LOOP
    fetch c_ventas into vdProPk, vdUbiPk, vdTiempoPk, v_cantidad, v_ingresos;
    exit when c_ventas%NOTFOUND;
    insert into h_ventas (hventaspk,dprodpk,dubipk,dtiempopk,cantidad,ingresos)
    values (seq_h_ventas.nextval, vdProPk, vdUbiPk, vdTiempoPk, v_cantidad, v_ingresos);
    commit;
  END LOOP;
  close c_ventas;
END ACTUALIZA_VENTAS;
/
```

Ejecuta el procedimiento ACTUALIZA_VENTAS para poblar con datos entre las fechas **1 de enero de 2022** y **18 de marzo de 2022**.

EVIDENCIA CAPTURA DE PANTALLA EN DONDE SE MUESTRE EL TOTAL DE REGISTROS DE LA TABLA DE HECHOS



Página de bienvenida | DWA01703613 | H_VENTAS | ACTUALIZA_VENTAS

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

	HVENTASPK	DPRODPK	DUBIPK	DTIEMPOPK	CANTIDAD	INGRESOS
2638	2930	6	1	66	42	1050
2639	2931	12	4	70	16	240
2640	2932	6	1	70	41	1025
2641	2933	9	3	72	75	1500
2642	2934	8	2	72	21	315
2643	2935	11	2	72	9	162
2644	2936	6	4	73	6	150
2645	2937	4	3	74	108	2376
2646	2938	4	1	75	37	814
2647	2939	5	4	9	14	322
2648	2940	3	1	9	22	286
2649	2941	1	3	10	58	2262
2650	2942	10	3	1	70	910
2651	2943	10	1	1	23	299
2652	2944	11	1	2	54	972
2653	2945	4	3	5	79	1738
2654	2946	9	2	7	6	120
2655	2947	3	2	8	41	533
2656	2948	6	4	17	18	450
2657	2949	9	2	18	10	200
2658	2950	10	4	24	14	182
2659	2951	3	2	12	15	195
2660	2952	11	4	25	17	306
2661	2953	10	4	26	48	624
2662	2954	12	3	15	22	330
2663	2955	1	3	30	35	1365
2664	2956	6	1	31	49	1225
2665	2957	9	2	32	19	380
2666	2958	12	2	27	30	450
2667	2959	9	4	36	5	100
2668	2960	9	1	36	11	220
2669	2961	9	2	38	20	400
2670	2962	5	3	39	30	690
2671	2963	8	4	39	19	285
2672	2964	5	3	46	68	1564
2673	2965	1	3	47	48	1872
2674	2966	4	4	48	6	132
2675	2967	9	4	49	45	900
2676	2968	4	3	49	9	198

```
create or replace PROCEDURE ACTUALIZA_VENTAS
(
  FECHAINICIAL IN DATE,
  FECHAFINAL IN DATE
)
AS
vFechaInicial date;
vFechaFinal date;
vdProPk number;
vdTiempoPk number;
vdUbiPk number;
v_cantidad number;
v_ingresos number;
cursor c_tiempo is
select dtiempopk
from dwa01703613.d_tiempo
where fecha BETWEEN vFechaInicial and vFechaFinal;
```

```
cursor c_ventas is
select DPO.dprodpk, DUB.dubipk, TI.dtiempopk, SUM(VE.cantidad), SUM(VE.cantidad * VE.precio)
from a01703613.ventas VE, dwa01703613.d_producto DPO, dwa01703613.d_tiempo TI,
dwa01703613.d_ubicacion DUB
where VE.codproducto = DPO.codproducto
and VE.codsucursal = DUB.codsucursal
and TRUNC (VE.fechahora) = TI.fecha
group by DPO.dprodpk, DUB.dubipk, TI.dtiempopk;

BEGIN

  vFechaInicial := FECHAINICIAL;
  vFechaFinal := FECHAFINAL;

  open c_tiempo;
  LOOP
    fetch c_tiempo into vdTiempoPk;
    exit when c_tiempo%NOTFOUND;
    delete from dwa01703613.h_ventas where dtiempopk=vdTiempoPk;
    commit;
  END LOOP;
  close c_tiempo;

  open c_ventas;
  LOOP
    fetch c_ventas into vdProPk, vdUbiPk, vdTiempoPk, v_cantidad, v_ingresos;
    exit when c_ventas%NOTFOUND;
    insert into dwa01703613.h_ventas (hventaspk,dprodpk,dubipk,dtiempopk,cantidad,ingresos)
    values (dwa01703613.seq_h_ventas.nextval, vdProPk, vdUbiPk, vdTiempoPk, v_cantidad,
v_ingresos);
    commit;
  END LOOP;
  close c_ventas;

END ACTUALIZA_VENTAS;
```

Parte V Consultas

La siguiente consulta nos da como resultado las ventas por producto acumulado por mes desde el 1 de enero hasta el 1 de abril del 2022. Impleméntala en una vista que se llame ventaProductoMes

```
select descproducto as "PRODUCTO",
sum(cantidad) as "CANTIDAD VENDIDA",
MES
from d_producto dp,
h_ventas hv,
d_tiempo dt
where dp.dprodpk = hv.dprodpk
and dt.dtiempopk = hv.dtiempopk
and dt.fecha between to_date ('01-01-2022','DD-MM-YYYY') and to_date('01-04-2022','DD-MM-YYYY')
group by descproducto, MEs
order by 1,to_char(to_date(MES,'Month'),'MM') asc
```

El resultado de la consulta les debe dar algo como esto:

	PRODUCTO	CANTIDAD VENDIDA	MES
1	Coca cola	3291	JANUARY
2	Coca cola	2751	FEBRUARY
3	Coca cola	1899	MARCH
4	Fritos	3353	JANUARY
5	Fritos	2912	FEBRUARY
6	Fritos	1897	MARCH
7	Manzana Lift	3500	JANUARY
8	Manzana Lift	2801	FEBRUARY
9	Manzana Lift	1986	MARCH
10	Pay de nuez	3416	JANUARY
11	Pay de nuez	2968	FEBRUARY
12	Pay de nuez	1736	MARCH
13	Pinguinos	3154	JANUARY
14	Pinguinos	3188	FEBRUARY
15	Pinguinos	1698	MARCH
16	Sabritas	3297	JANUARY
17	Sabritas	2973	FEBRUARY
18	Sabritas	1854	MARCH
19	Sabritones	3009	JANUARY
20	Sabritones	3203	FEBRUARY
21	Sabritones	1977	MARCH
22	Sprite	3623	JANUARY
23	Sprite	2679	FEBRUARY
24	Sprite	1610	MARCH
25	Submarinos	3156	JANUARY
26	Submarinos	2982	FEBRUARY
27	Submarinos	1864	MARCH
28	Victoria	3372	JANUARY
29	Victoria	2827	FEBRUARY
30	Victoria	1913	MARCH

Siguiendo el mismo ejemplo anterior implementen las vistas siguientes:

- Ventas de cada uno de los productos para todos los viernes del mes de FEBRERO
- Ventas de “Pay de Nuez” en cada sucursal
- Ventas de “Pinguinos” en cada sucursal, pero solamente los miércoles
- Ventas de TODOS los productos por sucursal y mes

EVIDENCIA

CAPTURA DE PANTALLA EN DONDE SE MUESTRE LA CONSULTA Y EL RESULTADO DE LAS MISMAS – 5 REPORTES – ANTERIORES.

```sql
-- Ventas de cada uno de los productos para todos los viernes del mes de FEBRERO

select descproducto as "PRODUCTO",sum(cantidad) as "CANTIDAD VENDIDA"
from d_producto dp,h_ventas hv,d_tiempo dt,d_ubicacion du
where dp.dprodpk = hv.dprodpk and dt.dtiempopk =hv.dtiempopk and du.dubipk=hv.dubipk
and dt.fecha between to_date ('01-02-2022','DD-MM-YYYY') and to_date('28-02-2022','DD-MM-
YYYY')
and dt.nombredia ='FRIDAY'
group by descproducto
order by 1 asc;
```

PRODUCTO	CANTIDAD VENDIDA
Coca cola	455
Fritos	432
Manzana Lift	388
Pay de nuez	414
Pinguinos	434
Sabritas	363
Sabritones	525
Sprite	413
Submarinos	410
Victoria	429

10 filas seleccionadas.

```sql
-- Ventas de “Pay de Nuez” en cada sucursal

select descproducto as "PRODUCTO",du.codsucursal,sum(cantidad) as "CANTIDAD VENDIDA"
from d_producto dp,h_ventas hv,d_tiempo dt,d_ubicacion du
where dp.dprodpk = hv.dprodpk and dt.dtiempopk =hv.dtiempopk and du.dubipk=hv.dubipk
and dt.fecha between to_date ('01-01-2022','DD-MM-YYYY') and to_date('01-04-2022','DD-MM-
YYYY')
and dp.descproducto ='Pay de nuez'
group by descproducto, du.codsucursal
order by 1 asc;
```

PRODUCTO	CODSU	CANTIDAD VENDIDA
Pay de nuez	QRO	1940
Pay de nuez	MEX	1052
Pay de nuez	MTY	3615
Pay de nuez	GDA	1513

```sql
--Ventas de “Pinguinos” en cada sucursal, pero solamente los miércoles

select descproducto as "PRODUCTO",du.codsucursal,sum(cantidad) as "CANTIDAD VENDIDA"
from d_producto dp,h_ventas hv,d_tiempo dt,d_ubicacion du
where dp.dprodpk = hv.dprodpk and dt.dtiempopk =hv.dtiempopk and du.dubipk=hv.dubipk
and dt.fecha between to_date ('01-01-2022','DD-MM-YYYY') and to_date('01-04-2022','DD-MM-
YYYY')
and dp.descproducto ='Pinguinos' and dt.nombredia ='WEDNESDAY'
group by descproducto, du.codsucursal
order by 1 asc;
```

```
PRODUCTO  CODSU CANTIDAD VENDIDA
--------- ----- ----------------
Pinguinos MEX                270
Pinguinos QRO                266
Pinguinos MTY                561
Pinguinos GDA                289
```

```sql
--Ventas de TODOS los productos por sucursal y mes
select descproducto as "PRODUCTO",du.codsucursal,sum(cantidad) as "CANTIDAD VENDIDA"
from d_producto dp,h_ventas hv,d_tiempo dt,d_ubicacion du
where dp.dprodpk = hv.dprodpk and dt.dtiempopk =hv.dtiempopk and du.dubipk=hv.dubipk
and dt.fecha between to_date ('01-01-2022','DD-MM-YYYY') and to_date('01-04-2022','DD-MM-
YYYY')
group by descproducto, du.codsucursal
order by 1 asc;
```

```
PRODUCTO                                           CODSU CANTIDAD VENDIDA
-------------------------------------------------- ----- ----------------
Coca cola                                          GDA              1659
Coca cola                                          MEX              1087
Coca cola                                          MTY              3415
Coca cola                                          QRO              1780
Fritos                                             GDA              1581
Fritos                                             MEX              1082
Fritos                                             MTY              3406
Fritos                                             QRO              2093
Manzana Lift                                       GDA              1656
Manzana Lift                                       MEX              1062
Manzana Lift                                       MTY              3560


Manzana Lift                                       QRO              2009
Pay de nuez                                        GDA              1513
Pay de nuez                                        MEX              1052
Pay de nuez                                        MTY              3615
Pay de nuez                                        QRO              1940
Pinguinos                                          GDA              1504
Pinguinos                                          MEX              1246
Pinguinos                                          MTY              3090
Pinguinos                                          QRO              2200
Sabritas                                           GDA              1414
Sabritas                                           MEX              1022

PRODUCTO                                           CODSU CANTIDAD VENDIDA
-------------------------------------------------- ----- ----------------
Sabritas                                           MTY              3607
Sabritas                                           QRO              2081
Sabritones                                         GDA              1572
Sabritones                                         MEX              1051
Sabritones                                         MTY              3417
Sabritones                                         QRO              2149
Sprite                                             GDA              1556
```