Base de Datos 2:
Proyecto1

Estudiantes:
Elias Santiago
Ronald Mariotti
Edward Pimentel
Regis Baez

Matricula:
2014-0717
2014-0698
2014-0909
2014-0324

Profesor:
Prof. Máximo E. Pérez M.

INTRODUCCION

En este proyecto se pide realizar un proceso ETL que nos sirva para limpiar, pulir y estandarizar los datos de las tablas para poder llevar un modelo operacional (previamente pensado, diseñado y hecho para poder moverlo con el ETL) a un modelo de dimensiones que se pueda usar para poder tener una buena base de datos que nos pueda servir para poder analizar las carreras de manera rigurosa y específica, ya que con esto se obtienes mejores análisis que nos sirven para tomar mejores decisiones sobre las carreras, en este caso es el desempeño de los corredores que se mide con el tiempo tomado de cada corredor, también debemos hacer el ETL para poder cumplir con cada una de las descripciones que hacen una buena Data WareHouse que debe de servir tener los datos de una manera legible, organizada y fácil de acceder.

El proceso seguido para este Proyecto consiste en lo siguiente:

-El diseño del modelo de datos operacional, basado en las especificaciones requeridas y los modelos previos

-El diseño del modelo dimensional, basado en el modelo relacional

-Elaboración física de ambos modelos

-El proceso de ETL

-Realizar las consultas requeridas/deseadas

El proceso a modelar en este proyecto es el de una carrera. Al principio del semestre, el curso realizo una carrera experimental como ejercicio para tener una idea del tipo de datos que tendríamos que recolectar de una actividad similar. Para esta actividad se recopilo información sobre los participantes como su peso, estatura, y edad entre otros datos útiles. Luego se realizó el recorrido en la pista de 400 metros, y se tomó el tiempo para calcular la velocidad. Modelamos nuestra base de datos en base a esto, pero le agregamos más complejidad para poder manejar una mayor cantidad de datos, en adición a los elementos planteados en los requerimientos del proyecto.

Modelo detallado.

Las ventajas de este modelo, además de tener más información, es que los tiempos se guardan en tramos definidos cada cierta distancia. Los tramos pueden tener poca o mucha distancia entre ellos, dependiendo del evento que sea y cuantas medidas de tiempo se quiera. De los tramos se puede definir la inclinación, esto es útil para cuando sea un evento donde la pista sea variable, como una carrera de mountain bike, y también el material del tramo, por si el evento es un triatlón y se cambie completamente el terreno. Esa es básicamente la ventaja principal del modelo más detallado, otras cosas que este modelo toma en cuenta es que divide los países en regiones y estas en sectores, para detallar más la localización del terreno del evento.

La desventaja principal seria la cantidad de medidas que se tuvieran que tomar para medir el tiempo, dependiendo de la distancia entre los tramos, también que puede haber errores si estos datos se toman de manera manual.

Modelo poco detallado.

La ventaja de este modelo es que los datos serían más simples y concisos, hubiera menos riesgos de poner datos erróneos. También sería más rápido insertar datos y se procesarían más rápido. Las ventajas de este modelo simplemente serian cuestiones de velocidad.

Entre las desventajas de este modelo están las ventajas del modelo más detallado, en este modelo solo se toman los tiempos de inicio y final, lo que limitaría el análisis de rendimiento del corredor, también no se puede analizar los distintos terrenos e inclinación variable. Los terrenos se limitarían a un solo material por lo cual no se pudieran analizar eventos donde el terreno varia por pedazos de la carrera.

Descripción general del proceso de ETL.

Nuestro grupo comenzó por tener completo el diseño físico de ambas bases de datos, la operacional y la dimensional. Después de tener esta parte fundamental, y después de tener datos presentes en la base de datos operacional, empezamos a transformar (preparar los datos para la base de datos dimensional, con joins necesarios entre otros procesos) y cargar los datos a esta base de datos. Luego de esto se comenzaron a crear los querys para realizar el proceso de carga del ETL.

Iniciamos las pruebas tratando de usar MERGE para cargar los datos de un schema a otro, pero luego nos dimos cuenta de que había varias dificultades dentro de postgres a la hora de usar MERGE, ya que este no está implementado dentro del SQL que usa postgres.

Decidimos entonces, realizar un proceso más simple para cargar los datos haciendo uso de un query de SQL básico. El INSERT, se inició el proceso de carga de datos a base de inserts para cada una de las tablas del modelo dimensional desde el operacional. Tuvimos varios inconvenientes a la hora de cargar los datos, nos dimos cuenta de algunos fallos dentro de nuestro modelo operacional y dimensional.

En medio del proceso de cargar, se realizaron diferentes cambios a ambos modelos para resolver los inconvenientes que se presentaron (problemas con algunos FK, valores NULL y NOT NULL). Además, nos dimos cuenta de la necesidad de crear una view o tabla temporal para cargar el fact table. Eso trajo consigo nuevos problemas y modificaciones.

No lográbamos hacer que funcionen los UPDATE debido al problema que se presentó con nuestro fact table. Tratamos con diferentes soluciones y la tabla intermedia, pero el error no cesaba y la carga datos no era realizada hacia el fact table.

Como decisión final tratamos de asegurarnos que la base de datos se mantenga los más estable y consistente posible, verificamos cada entrada y salida de datos que fueron cargadas a las tablas dimensionales. Mientras que el fact table al menos mantuviera una buena estructura.

Descripciones cualitativas.

El proceso de ETL (Extraction-Transform-Load), ha resultado ser bastante útil para realizar la migración de una base de datos completa en muy poco tiempo y con la capacidad de mejorar el rendimiento de la misma.

Al iniciar el proceso ETL nos dimos cuenta de lo tedioso que podía ser el traspase o migración de datos de una base de datos a otra. Debido que hubiésemos tenido que reconstruir desde cero la base de datos completa para poder traspasar todos los datos que se encontraban en ella.

Para una compañía, el tener que hacer todo desde cero cuesta bastante dinero y tiempo. En esto pudimos ver que el proceso de ETL, no solo ahorra bastante tiempo para migrar datos, sino que también el costo de esfuerzo que conlleva.

Entre otras cualidades que encontramos, el ETL puede ayudar a mejorar el rendimiento y carga de la base de datos.

Por lo tanto, es un método muy útil que puede ser aplicado tanto en grandes compañías como un negocio pequeño y ayuda a mejorar el rendimiento de la misma.

Conclusión

En fin, se puede ver de la utilidad e importancia que tiene el proceso del ETL ya que con el uno puede llevar del modelo operacional hacia el modelo dimensional que es mucho mejor para poder apreciar analizar y observar todos los datos, y de ahí se pueden analizar los procesos más detallados y se puede lograr una mejor toma de decisiones.

Además de ser muy útil para agilizar la carga de datos de una base de datos y mejorar su rendimiento para una mejor funcionalidad.

CREATE TABLES y ETL QUERYS.

“L” – ETL

```
INSERT INTO dim_model."Dim_Persona" ("ID_persona",
        "Prim_Nombre",
        "Seg_Nombre",
        "Prim_Apellido",
        "Seg_Apellido",
        "Fecha_Nacimiento",
        "Sexo")
   SELECT "id_per",
        "Primer_nombre",
        "Segundo_nombre",
        "Primer_Apellido",
        "Segundo_Apellido",
        "Fecha_Nacimiento",
        "Sexo" FROM public."Persona"
```

```
        INSERT INTO dim_model."Dim_Atleta" ("ID_atleta",
        "Porcentaje_grasa" ,
        "Porcentaje_agua" ,
        "Masa_muscular" ,
```

```
        "Grasa_visceral" ,
        "Peso" ,
        "Altura" ,
        "Persona_id" ,
        "Entrenador_id"

)
SELECT "id_atleta",
"Porciento_grasa",
        "Porciento_agua",
        "Masa_Muscular",
        "Grasa_visceral",
        "Peso",
        "Altura",
        "id_persona",
"Entrenador"
FROM public."Atleta"



        --------------------------------

        INSERT INTO dim_model."Dim_Entrenador" ("id_entrenador" ,
        "Persona_id_entrenad",
        "Fecha_ingreso")

SELECT "id_entrenador" ,
        "id_persona",
        "fecha_ingreso"
FROM public."Entrenador"
```

```
	----------------------------------


		INSERT INTO dim_model."Dim_Clima" (
"ID_clima" ,
	"Estacion",
	"Temperatura" ,
	"Precipitacion" ,
	"Velocidad_viento",
	"Porcentaje_humedad",
	"Fecha_clima" ,
	"Direccion_viento"

)
SELECT "id_clima" ,
	"Estacion",
	"Temperatura" ,
	"Precipitacion" ,
	"Velocidad_viento",
"Porcentaje_humedad",
	"Fecha_clima" ,
"Direccion_viento"

FROM public."Clima"

	---------------------------------------
```

```
                INSERT INTO dim_model."Dim_Estacion" (
"ID_estacion",
        "Nombre"

)
SELECT "id_estacion" ,
        "Nombre"

FROM public."Estacion"

        -----------------------------------

                INSERT INTO dim_model."Dim_Dirrecion_Viento" (
"ID_viento" ,
        "Direccion"

)
SELECT "id_direccion" ,
        "Direccion"

FROM public."Direccion"

        -----------------------------------

                INSERT INTO dim_model."Dim_Material" (
"id_material" ,
        "Tipo" ,
        "Nombre"

)
```

```
SELECT  "id_material" ,
	"Tipo" ,
	"Nombre"

FROM public."Material"

	---------------------------------------



		INSERT INTO dim_model."Dim_Hora" (
	"ID_hora" ,
	"Hora" ,
	"Minutos" ,
	"Segundos"

)
SELECT
"id_tiempo" ,
	"Hora" ,
	"Minutos" ,
	"Segundos"

FROM public."Tiempo"

	----------------------------------------



	INSERT INTO dim_model."Dim_Tramo" ("id_tramo",
				"Inclinacion",
				"Material")
```

```
SELECT "id_tramo",
"Inclinacion",
"Material"

FROM public."Tramo"

	----------------------------------------


	INSERT INTO dim_model."Dim_Provincia" ("id_provin",

				"Nombre")

SELECT "id_provincia",

"Nombre"

FROM public."Provincia"

	----------------------------------------


	INSERT INTO dim_model."Dim_Provincia" ("id_provin",

				"Nombre")

SELECT "id_provincia",

"Nombre"
```

```
FROM public."Provincia"

	-----------------------------------------



	INSERT INTO dim_model."Dim_Region" ("id_reg",

				"Nombre")

SELECT "id_region",

"Nombre"

FROM public."Region"

	-------------------------------------------



	INSERT INTO dim_model."Dim_Pais" ("id_pais",
				"Nombre",
			"Region")

SELECT "id_paises",
"Nombre",
"id_regiones"
```

```
-----------------------------------------------------------------

FROM  public."Pais_Region" as PR,public."Pais" as Pa, public."Region" as R

where PR."id_paises"=Pa."id_pais" and PR."id_regiones"=R."id_region"

	-------------------------------------------

	INSERT INTO dim_model."Dim_Sector" ("id_sect",
					"Nombre",
				       "Provincia")

SELECT "id_sector",
"Nombre",
"Provincia"



FROM  public."Sector"

	-----------------------------------------

	INSERT INTO dim_model."Dim_Terreno" ("ID_terreno",
					"Altitud",
```

```
                        "Sector_id",
                          "Inclinacion",
                          "Material_terreno",
                          "Condicion_terreno"
                        )

SELECT "id_terreno",
"Altitud_sob_mar",
"Sector",
"Inclinacion",
"Material",
"Condicion"

FROM  public."Terreno"


      ---------------------------------------------

      create view Load_Fact_Table as select(       "Clima_id",
                                  "Terreno_id",
                                   "Atleta_id",
                                   "Evento_id",
                              "Distancias",
                              "vuelta",
                              "hora_tramo",
                              "minuto_tramo",
                              "segundo_tram",
                              "milisegundo_tramo",
                              "Vuelta_Max"
                        )
```

from public."Persona" as PP,public."Terreno" as TE,public."Corrida" as CO,public."Evento" as EV,public."Tramo" as TR,public."Clima" as CL,public."Tramo_Corrida" as TC, public."Atleta" as Atl

where Clima_id=id_clima;

and Terreno_id=id_terreno;

and Atleta_id=id_atleta;

and Evento_id=id_evento;

and Distancias=Distancia

Queries:

--1. **Evento_id mas popular.**

select Evento_id, count(Evento_id) as cantidad_participantes from corrida group by Evento_id order by count(Evento_id);

--2. **Relacion entre el tiempo total y la altura.**

select (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)), altura from corrida order by (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)) asc;

--3. **Relacion entre el tiempo total y el peso.**

select (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)), peso from corrida order by (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)) asc;

--4. **Relacion entre el tiempo total y el IMC.**

Select sum(tiempo_tramo), TRUNC((peso*0.45)/((altura*0.02)^2),1) as IMC from corrida order by (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)) asc;

--5 **Promedio del estado fisico de los corredores.**

select avg(peso) as promPeso, trunc(avg(altura),1) as promAltura, trunc(avg(grasa),1) as promGrasa, avg(grasa_visceral) as promGrasaVisceral, trunc(avg(masa_muscular),1) as promMusculo from corrida;

--6 **Porcentaje de agua optimo.**

select agua, (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)) from corrida order by (sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo)) asc;

--7 **Condicion del Terreno_id donde se promedia mejor.**

select material_Terreno_id, altitud, inclinacion, Condicion_Terreno_id, trunc(max(promTiempo),2) as tiempoPromediado from (select avg((sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo))) as promTiempo from corrida) as sub, Terreno_id group by material_Terreno_id, altitud, inclinacion, Condicion_Terreno_id;

--8 **Atleta femenino que ha participado mas.**

select max(participaciones) as maxParticipaciones, id_persona, Atleta_id, sexo from (select count(Atleta_id) as participaciones from corrida) as sub, corrida, personas where Atleta_id = id_persona and sexo = 'f' group by Atleta_id, id_persona

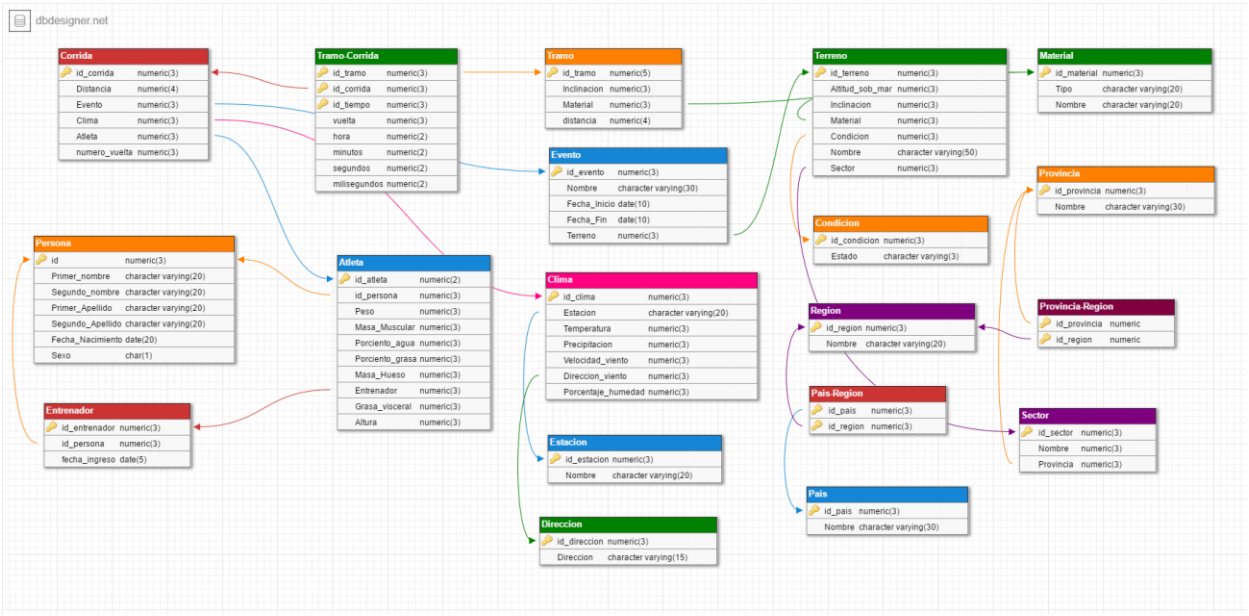
--9 **Velocidad promedio de una corrida de un Evento_id determinado.**

select trunc(avg(distancia*vueltas/(sum(hora_tramo)+sum(minuto_tramo)+sum(segundo_tramo)+sum(milisegundo_tramo))),2) as VelocidadPromedio, Evento_id from corrida

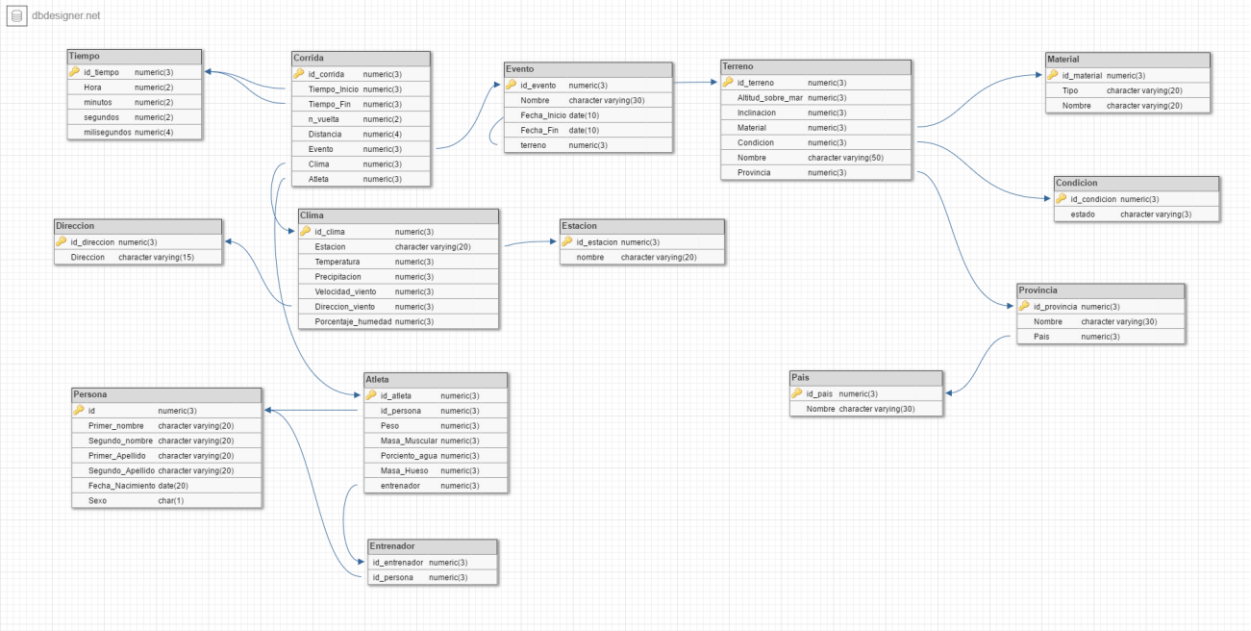
group by Evento_id

Modelos:

Modelo detallado.



Modelo no detallado.



Modelo dimensional.