

UNIVERSIDADE FEDERAL DO PARANÁ

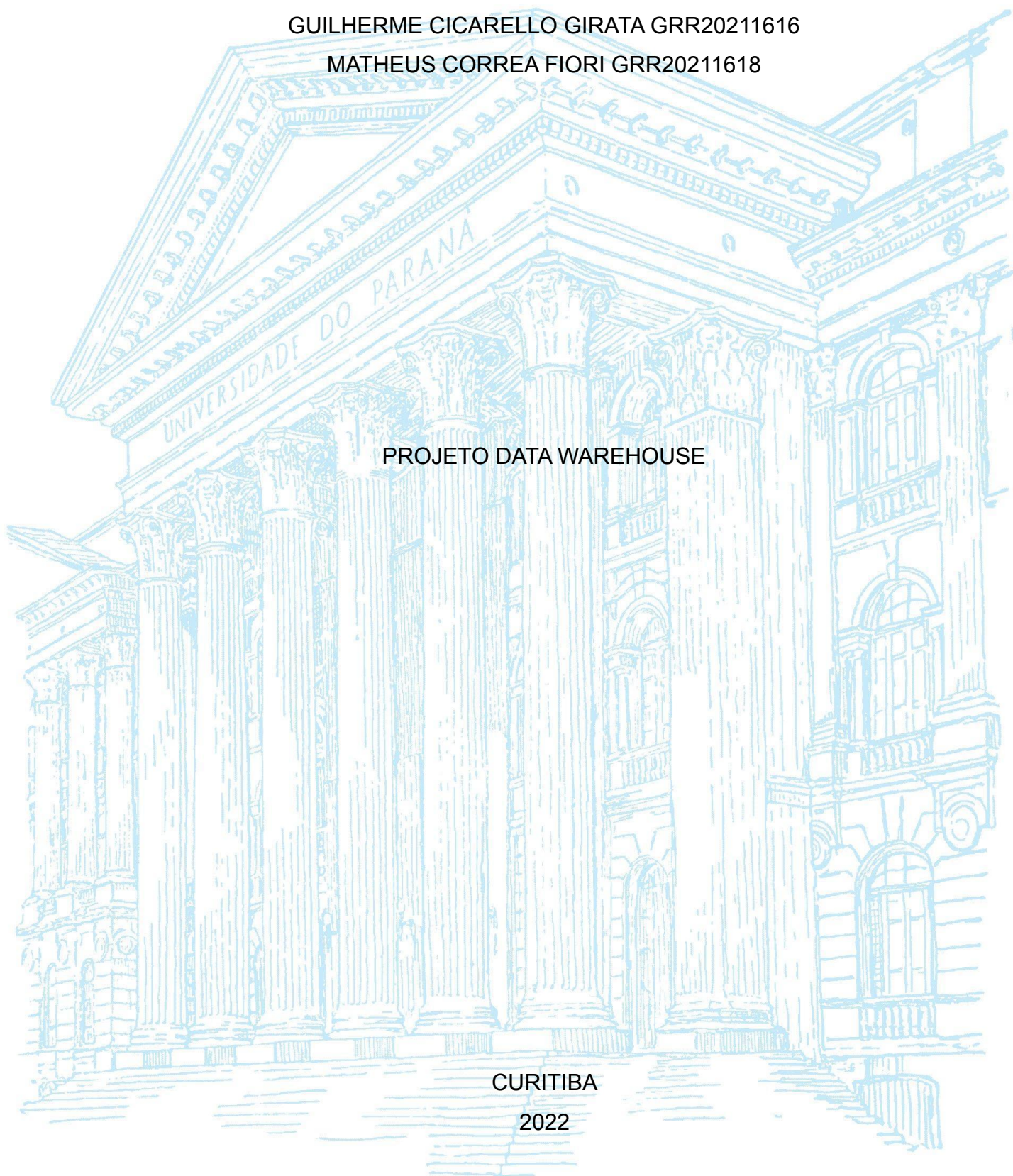
GUILHERME CICARELLO GIRATA GRR20211616

MATHEUS CORREA FIORI GRR20211618

PROJETO DATA WAREHOUSE

CURITIBA

2022



1.Introdução

Após o final da pandemia do COVID-19, a loja de varejo GA, uma franquia aberta por três sócios, teve um ano muito produtivo e lucrativo e está procurando expandir os seus negócios e operações. O objetivo dos sócios é abrir novas filiais na cidade de Curitiba e, posteriormente, expandir para novas cidades, na medida do possível.

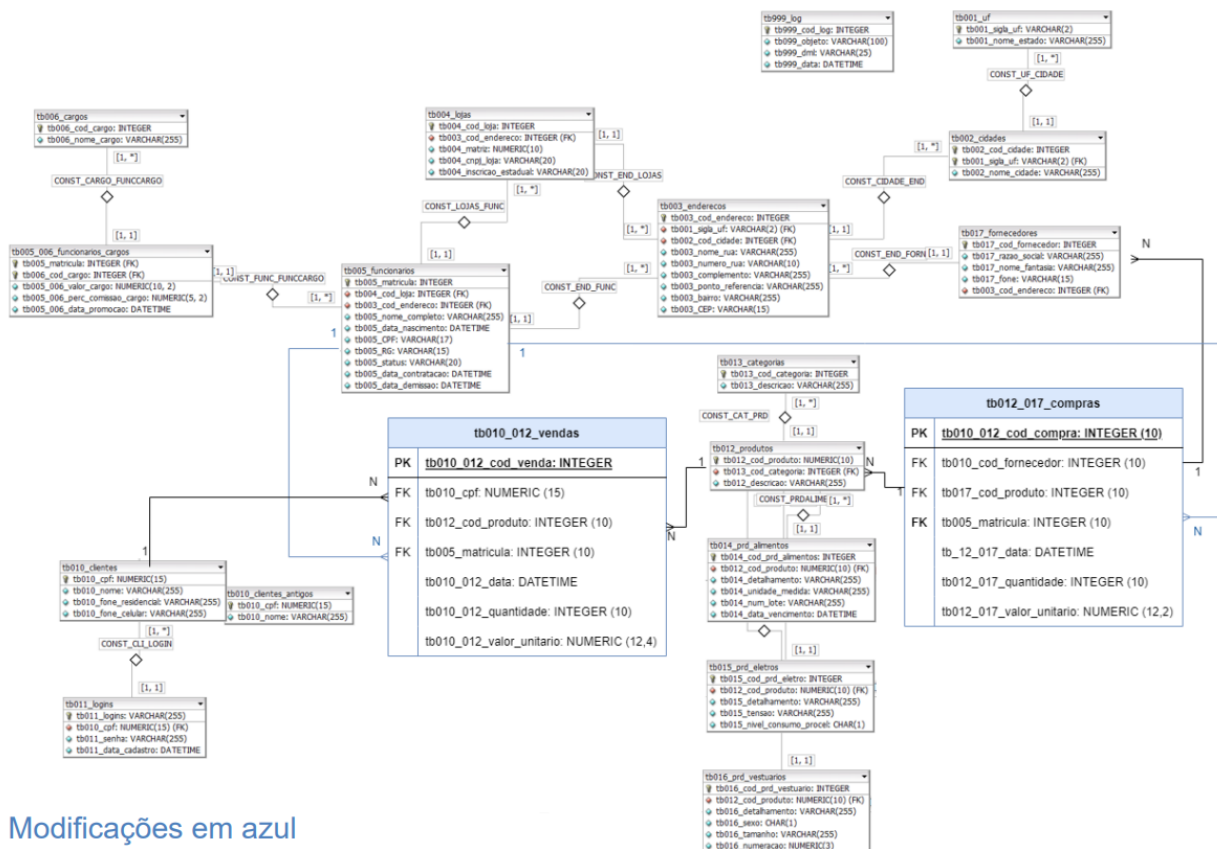
Com o rápido crescimento da empresa, até o momento regional, os sócios têm tido problemas com os estoques das lojas e com os funcionários. A loja vem recebendo duras críticas de gerentes e clientes, por falhas no planejamento e organização dos estoques, e pelo grande número de faltas não justificadas dos funcionários. Esses fatos vem afetando grandemente os negócios da empresa.

Os sócios precisam procurar uma maneira efetiva para organizar os estoques e os itens vendidos, assim como garantir a permanência dos funcionários durante o expediente. Um dos sócios sugere, também, a criação de um histórico de vendas para rastrear e calcular o rendimento de cada filial, permitindo melhor análise individual de cada sede e sua respectiva área mais rentável.

Após uma conversa sobre a situação os sócios chegaram a conclusão que, para ter uma melhor organização de estoque e vendas será necessário a implementação de um Data Warehouse para a loja. Eles acreditam que o sistema deve ser capaz de registrar todos os dados relacionados à loja, sejam eles produtos, funcionários, clientes, endereços, fornecedores, etc. Assim como, resolver os problemas discutidos anteriormente, como os estoques mal planejados e fora de organização, deixando-os separados por categorias facilitando o acesso para todos.

Dessa maneira, as lojas, funcionários e clientes devem possuir um ID único para manter a organização e rastreamento. Cada venda efetuada será vinculada a um vendedor, assim como as compras terão todos os itens registrados com o preço total e individual.

2. Situação e Modelo Atual



Modificações em azul

2.1 Descrição

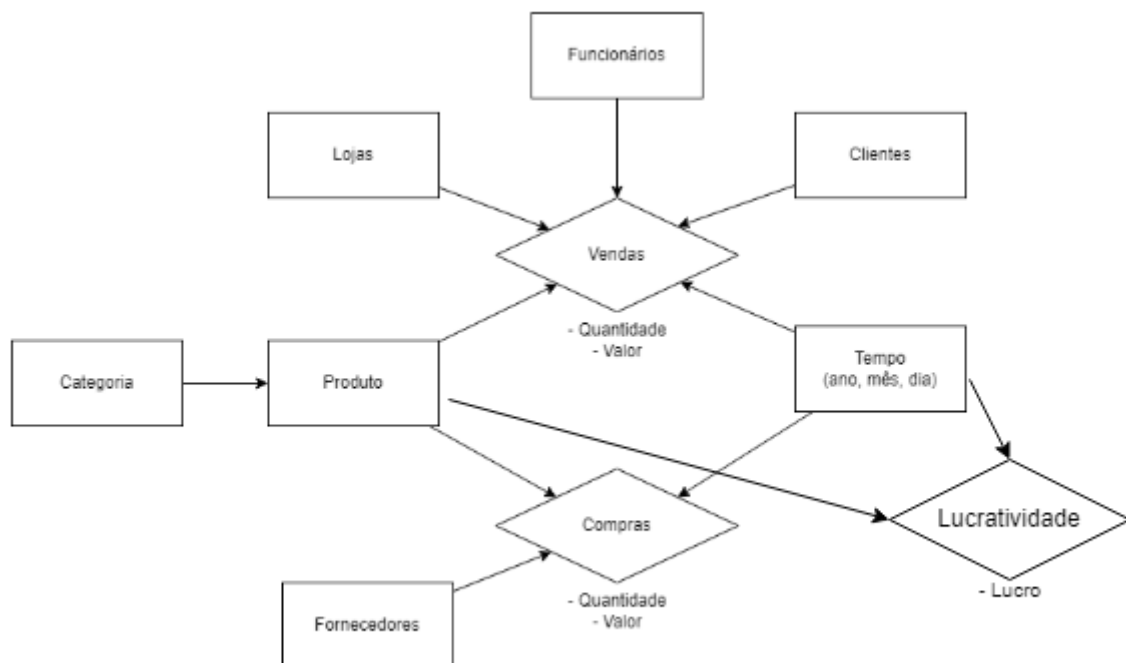
No diagrama acima as tabelas de “vendas” e “itens_vendas” foram agregadas em uma só tabela, ocorrendo a normalização dos dados de ambas as tabelas. Assim como na tabela “compras” e “itens_compras”. O resto das tabelas se manteve igual. A associação entre vendas, produtos e compras é feita de forma em que é possível fazer sua visualização facilmente, assim, as tabelas vendas e compras possuem “cod_produto” como chave estrangeira da tabela produtos.

3. Data Warehouse Proposto

3.1 Descrição e Justificativa

Para conseguir atender as expectativas e necessidades da loja de varejo, o Data Warehouse propõe um tratamento de dados, que terá como foco trazer resultados e uma possível melhora na qualidade da análise dos dados, visando a entrega de melhores resultados e melhor análise gerencial pelos diretores. Nos tópicos consequentes, será possível compreender melhor a estrutura e funcionamento do Data Warehouse.

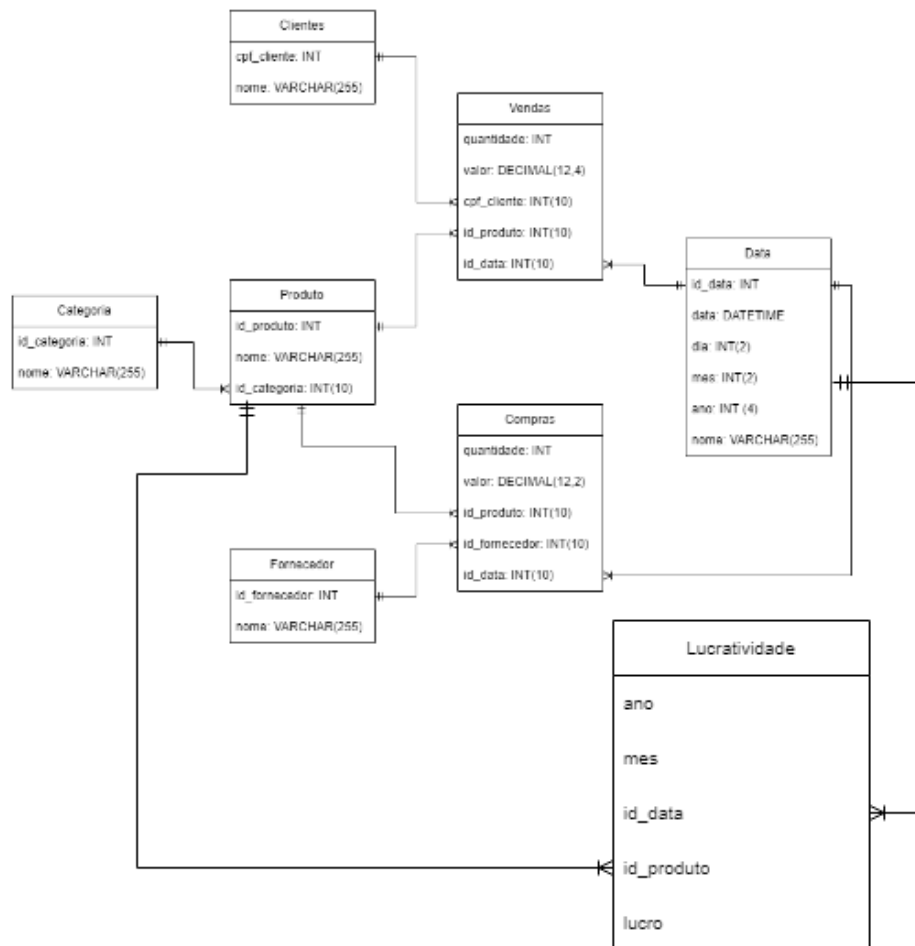
4. Modelo Conceitual



4.1 Descrição do modelo conceitual

O modelo conceitual do Data Warehouse da loja, evidencia dois fatos, o fato “vendas” e o fato “compras” que possuem como medidas “quantidade” e “valor”. Ligados ao fato “vendas”, temos as dimensões: lojas, funcionários e clientes, assim como as dimensões produtos e tempo, que por sua vez também fazem ligação com o fato “compras”, que possui ligação independente com a dimensão fornecedores. O fato lucratividade, tem como medida o lucro, recebendo como dimensões o tempo e produto.

5. Modelo Lógico



5.1 Descrição do modelo lógico

O modelo lógico do Data Warehouse da loja, os fatos “vendas”, “compras” e “lucratividade” recebem as chaves estrangeiras de outras tabelas (categoria, produto, fornecedor, funcionário, data, vendas e clientes), garantindo unicidade e abstração dos dados, além de ficar mais claro como as tabelas se relacionam e quais são os dados e atributos que cada tabela possui.

6. ETL's

```
CREATE TRIGGER Trigger_Clientes AFTER INSERT
ON datawarehouse.tb010_clientes
FOR EACH ROW
BEGIN
```

```
    INSERT INTO DW.Clientes VALUES (new.tb010_cpf, new.tb010_nome, null);
END;
```

```
CREATE TRIGGER Trigger_ClientesVenda AFTER INSERT
ON datawarehouse.tb010_012_vendas
FOR EACH ROW
BEGIN
```

```
    UPDATE DW.Clientes SET ultima_venda = new.tb010_012_data WHERE cpf_cliente =
new.tb010_cpf;
END
```

```
CREATE PROCEDURE Vendas_Mensais()
BEGIN
```

```
    INSERT INTO DW.Fato_Vendas (quantidade, id_produto, id_categoria, dia, dia_semana,
mes, ano)
```

```
    SELECT sum(v.tb010_012_quantidade), v.tb012_cod_produto, p.tb013_cod_categoria,
day(v.tb010_012_data), weekday(v.tb010_012_data), month(v.tb010_012_data),
year(v.tb010_012_data)
```

```
    FROM datawarehouse.tb010_012_vendas V, datawarehouse.tb012_produtos P,
datawarehouse.tb005_funcionarios F, datawarehouse.tb004_lojas L
```

```
    WHERE p.tb012_cod_produto = v.tb012_cod_produto AND v.tb005_matricula =
f.tb005_matricula AND f.tb004_cod_loja = l.tb004_cod_loja AND v.tb010_012_data BETWEEN
DATE_SUB(NOW(), INTERVAL 1 MONTH) AND NOW()
```

```
    GROUP BY v.tb012_cod_produto, v.tb010_012_data, l.tb004_cod_loja;
```

```
    INSERT INTO DW.Fato_Vendas (quantidade, id_produto, id_categoria, dia, dia_semana,
mes, ano)
```

```
    SELECT sum(v.tb010_012_quantidade), v.tb012_cod_produto, p.tb013_cod_categoria, null,
weekday(v.tb010_012_data), month(v.tb010_012_data), year(v.tb010_012_data)
```

```
    FROM datawarehouse.tb010_012_vendas V, datawarehouse.tb012_produtos P,
datawarehouse.tb005_funcionarios F, datawarehouse.tb004_lojas L
```

```
    WHERE p.tb012_cod_produto = v.tb012_cod_produto AND v.tb005_matricula =
f.tb005_matricula AND f.tb004_cod_loja = l.tb004_cod_loja AND v.tb010_012_data BETWEEN
DATE_SUB(NOW(), INTERVAL 1 MONTH) AND NOW()
```

```
    GROUP BY v.tb012_cod_produto, weekday(v.tb010_012_data), month(v.tb010_012_data),
year(v.tb010_012_data), l.tb004_cod_loja;
```

```
    INSERT INTO DW.Fato_Vendas (quantidade, id_produto, id_categoria, dia, dia_semana,
mes, ano)
```

```
    SELECT sum(v.tb010_012_quantidade), v.tb012_cod_produto, p.tb013_cod_categoria, null, null,
month(v.tb010_012_data), select year(v.tb010_012_data)
```

```
    FROM datawarehouse.tb010_012_vendas V, datawarehouse.tb012_produtos P,
datawarehouse.tb005_funcionarios F, datawarehouse.tb004_lojas L
```

```

        WHERE p.tb012_cod_produto = v.tb012_cod_produto AND v.tb005_matricula =
f.tb005_matricula AND f.tb004_cod_loja = l.tb004_cod_loja AND v.tb010_012_data BETWEEN
DATE_SUB(NOW(), INTERVAL 1 MONTH) AND NOW()
        GROUP BY v.tb012_cod_produto, month(v.tb010_012_data), year(v.tb010_012_data),
l.tb004_cod_loja;
END

```

```

CREATE PROCEDURE Compras_Mensais()
BEGIN
    INSERT INTO DW.Fato_Compras (quantidade, valor, id_produto, mes, ano)
    SELECT sum(c.tb012_017_quantidade), sum(c.tb012_017_valor_unitario), c.tb012_cod_produto,
month(c.tb012_017_data), year(c.tb012_017_data)
    FROM datawarehouse.tb012_017_compras C
    WHERE c.tb012_017_data BETWEEN DATE_SUB(NOW(), INTERVAL 1 MONTH) AND NOW()
    GROUP BY c.tb012_cod_produto;
END ;

```

```

CREATE PROCEDURE Venda_Compra_Anual()
BEGIN
    INSERT INTO DW.Fato_Vendas (quantidade, id_produto, id_categoria, dia, dia_semana,
mes, ano)
    SELECT sum(v.tb010_012_quantidade), v.tb012_cod_produto, p.tb013_cod_categoria, null, null,
null, year(v.tb010_012_data)
    FROM datawarehouse.tb010_012_vendas V, datawarehouse.tb012_produtos P,
datawarehouse.tb005_funcionarios F, datawarehouse.tb004_lojas L
    WHERE p.tb012_cod_produto = v.tb012_cod_produto AND v.tb005_matricula =
f.tb005_matricula AND f.tb004_cod_loja = l.tb004_cod_loja AND v.tb010_012_data BETWEEN
DATE_SUB(NOW(), INTERVAL 1 YEAR) AND NOW()
    GROUP BY v.tb012_cod_produto, YEAR(v.tb010_012_data), l.tb004_cod_loja;

    INSERT INTO DW.Fato_Compras (quantidade, valor, id_produto, mes, ano)
    SELECT sum(c.tb012_017_quantidade), sum(c.tb012_017_valor_unitario), c.tb012_cod_produto,
null, year(c.tb012_017_data)
    FROM datawarehouse.tb012_017_compras C
    WHERE c.tb012_017_data BETWEEN DATE_SUB(NOW(), INTERVAL 1 YEAR) AND NOW()
    GROUP BY c.tb012_cod_produto, YEAR(c.tb012_017_data);
END

```

```

CREATE EVENT DW_mes
ON SCHEDULE EVERY '1' MONTH
STARTS '2023-03-10 00:00:00'
DO
BEGIN
    IF MONTH(NOW()) = 1 THEN
        CALL Venda_Compra_Anual();
    END IF;
    CALL Vendas_Mensais();
    CALL Compras_Mensais();
END

```

7. Frequência de Cargas

As cargas de dados serão realizadas diariamente para dentro do Data Warehouse.

8. Tempo de Retenção

Os dados serão armazenados no data warehouse por 7 anos. Porém, o usuário pode alterar o período de retenção se ele quiser manter esses dados por mais tempo ou então limpá-los de maneira mais agressiva.

9. Hardware Recomendado

Sistema operacional: Windows 7 (64-bits).

Processador: Intel Core 2 Quad CPU Q8400.

Memória RAM: 4,00 GB.

Armazenamento: 465 GB.

GPU: NVIDIA GeForce 210.

10. SGBD Recomendado

O SGBD recomendado para suportar a Data Warehouse realizada é o MySQL Server na versão 8.0.0.

12. Análise Gerencial

A Análise Gerencial será feita a partir de um procedure que será executado num intervalo de tempo de 15 dias, para que, a gerência da loja obtenha mais controle e inspeção sobre os dados fornecidos pelo DW.

12.1. Quantidade de produtos vendidos agrupados por categoria ao longo do tempo

```
select quantidade, id_produto, id_categoria, id_data, dia, dia_semana, mes, ano
from fato_vendas where id_produto is not null and id_categoria is not null and dia is
not null order by id_categoria;
```

quantidade	id_produto	id_categoria	id_data	dia	dia_semana	mes	ano
3	10	1	2	12	4	3	2010
1	12	1	8	18	3	3	2010
2	12	1	9	19	4	3	2010
2	14	1	14	24	2	3	2010
2	17	1	24	3	5	4	2010
1	11	1	6	16	1	3	2010
2	13	1	12	22	0	3	2010
1	15	1	18	28	6	3	2010
2	16	1	19	29	0	3	2010

12.2. Quantidade e valor total das compras por produto, possibilitando uma visão hierárquica ao longo do tempo

```
select quantidade, valor, id_produto, id_data, id_categoria, gasto_media, mes, ano
from fato_vendas where cpf_cliente is not null and valor is not null and gasto_media
is not null and quantidade is not null and mes is not null order by id_produto;
```

quantidade	valor	id_produto	id_data	id_categoria	gasto_media	mes	ano
5	1247.8000	30	29	3	38.42	4	2010
3	1259.9600	30	30	3	37.92	4	2010
2	1760.0000	31	31	3	58.67	4	2010
1	1600.0000	31	32	3	53.33	4	2010
1	577.7900	32	34	3	19.26	4	2010
1	574.4000	32	33	3	19.15	4	2010
1	894.4000	33	35	3	29.81	4	2010
1	737.7900	33	36	3	24.59	4	2010
2	49.6000	35	40	3	1.65	4	2010

12.3. Clientes que mais gastaram na loja virtual com quantidade acumulada, valor acumulado e média em um determinado período de tempo

```
select cpf_cliente, quantidade, valor, dia, mes, ano
from fato_vendas where cpf_cliente is not null and valor is not null and dia is not null
order by valor desc;
```

cpf_cliente	quantidade	valor	dia	mes	ano
10000000045	1	2949.7400	17	3	2010
10000000016	1	1968.9400	24	3	2010
10000000030	2	1760.0000	10	4	2010
10000000044	1	1643.3100	16	3	2010
10000000031	1	1600.0000	11	4	2010
10000000028	3	1152.6200	8	4	2010
10000000029	2	1137.5800	9	4	2010
10000000017	2	980.9100	25	3	2010
10000000034	1	894.4000	14	4	2010

12.4 Últimas compras realizadas por cliente, e tempo decorrido até a data atual

```
select * from fato_ultimacompra;
```

cpf_cliente	nome	Dia Ultima Venda
10000000000	NOME Teste 01	4675
10000000001	NOME Teste 02	4674
10000000002	NOME Teste 03	4673
10000000003	NOME Teste 04	4672
10000000004	NOME Teste 05	4671
10000000005	NOME Teste 06	4670
10000000006	NOME Teste 07	4669
10000000007	NOME Teste 08	4668

12.5 Lucratividade bruta dos produtos comprados e posteriormente vendidos

```
select * from fato_lucratividade where mes is not null order by lucro desc;
```

ano	mes	id_data	id_produto	lucro
2010	3	94	41	1106.15
2010	3	23	41	1106.15
2010	3	9	41	1106.15
2010	3	92	41	1106.15
2010	3	95	41	1106.15
2010	3	17	41	1106.15
2010	3	18	41	1106.15
2010	3	98	41	1106.15
2010	3	29	41	1106.15

12.6 Quantidades de atendimentos realizados ao longo do tempo

select quantidade, cpf_cliente, id_data, dia, dia_semana, mes, ano
from fato_vendas where matricula is not null and quantidade is not null and dia is not
null and dia_semana is not null;

quantidade	cpf_cliente	id_data	dia	dia_semana	mes	ano
1	100000000000	1	11	3	3	2010
1	100000000001	2	12	4	3	2010
1	100000000002	3	13	5	3	2010
1	100000000003	4	14	6	3	2010
1	100000000004	5	15	0	3	2010
2	100000000005	6	16	1	3	2010
1	100000000006	7	17	2	3	2010
1	100000000007	8	18	3	3	2010
1	100000000008	9	19	4	3	2010

13. Conclusão

Diante do problema que os sócios da loja de varejo estavam enfrentando, os mesmos tiveram a ideia de implementar um Data Warehouse para melhorar a eficácia do estoque e a gestão da empresa em geral.

Após a implementação do Data Warehouse os sócios ficaram muito satisfeitos com o resultado, pois finalmente foi possível organizar o estoque, separando-o por categorias, fornecedores e etc. Também ficou muito mais fácil controlar as vendas realizadas pela loja, podendo realizar inúmeras consultas no Banco de Dados que favorecem a obtenção de informações objetivas e cruciais para o desenvolvimento econômico da loja.

14. Referências

MARYNOWSKI, JOÃO. "UFPR Virtual - O Moodle Oficial: Log in to the Site." Ufprvirtual.ufpr.br, ufprvirtual.ufpr.br/course/view.php?id=27567. Accessed 18 Dec. 2022.

GIRATA, GUILHERME. "Notion – the All-In-One Workspace for Your Notes, Tasks, Wikis, and Databases." Notion, tidy-newsprint-e1c.notion.site/Banco-de-Dados-III-266638e9de374554bbd8fc624a1315bd. Accessed 22 Dec. 2022.

15. Anexos

```
CREATE DATABASE DW;  
USE DW;
```

```
CREATE TABLE Clientes(  
  cpf_cliente NUMERIC(15) primary key,  
  nome VARCHAR(255) not null  
);
```

```
CREATE TABLE Data(  
  id_data INTEGER primary key,  
  data DATETIME not null,  
  dia INTEGER not null,  
  mes INTEGER not null,  
  ano INTEGER not null,  
  nome VARCHAR(255) not null  
);
```

```
CREATE TABLE Categoria(  
  id_categoria INTEGER primary key,  
  nome VARCHAR(255) not null  
);
```

```
CREATE TABLE Produto(  
  id_produto NUMERIC(10) primary key,  
  nome VARCHAR(255) not null,  
  id_categoria INTEGER not null,  
  CONSTRAINT fk_Produto foreign key (id_categoria) REFERENCES Categoria(id_categoria)  
);
```

```
CREATE TABLE Fornecedor(  
  id_fornecedor INTEGER primary key,  
  nome VARCHAR(255) not null  
);
```

```
CREATE TABLE Funcionario(  
  matricula INTEGER primary key,  
  nome VARCHAR(255) not null  
);
```

```

CREATE TABLE Fato_Vendas(
quantidade INTEGER,
valor DECIMAL (12,4),
cpf_cliente NUMERIC(15),
id_produto NUMERIC(10),
id_data INTEGER,
CONSTRAINT fk_Vendas1 foreign key (cpf_cliente) REFERENCES Clientes(cpf_cliente),
CONSTRAINT fk_Vendas2 foreign key (id_produto) REFERENCES Produto(id_produto),
CONSTRAINT fk_Vendas3 foreign key (id_data) REFERENCES Data(id_data)
);

```

```

CREATE TABLE Fato_Compras(
quantidade INTEGER,
valor DECIMAL (12,2),
id_produto NUMERIC(10),
id_data INTEGER,
id_fornecedor INTEGER,
CONSTRAINT fk_Compras1 foreign key (id_produto) REFERENCES Produto(id_produto),
CONSTRAINT fk_Compras2 foreign key (id_data) REFERENCES Data(id_data),
CONSTRAINT fk_Compras3 foreign key (id_fornecedor) REFERENCES Fornecedor(id_fornecedor)
);

```

```

CREATE TABLE Fato_Lucratividade(
ano INTEGER,
mes INTEGER,
id_data INTEGER,
id_produto NUMERIC(10),
lucro NUMERIC(10,2),
CONSTRAINT fk_Lucratividade foreign key(id_data) REFERENCES Data(id_data),
CONSTRAINT fk_Lucratividade2 foreign key(id_produto) REFERENCES Produto(id_produto)
);

```

```
ALTER TABLE Clientes ADD COLUMN ultima_venda DATETIME;
```

```
ALTER TABLE Data DROP COLUMN nome;
```

```
ALTER TABLE Fato_Vendas ADD COLUMN id_categoria INTEGER;  
ALTER TABLE Fato_Vendas ADD CONSTRAINT fk_Vendas4 foreign key (id_categoria)  
REFERENCES Categoria(id_categoria);  
ALTER TABLE Fato_Vendas ADD COLUMN matricula INTEGER;  
ALTER TABLE Fato_Vendas ADD CONSTRAINT fk_Vendas5 foreign key (matricula)  
REFERENCES Funcionario(matricula);  
ALTER TABLE Fato_Vendas ADD COLUMN gasto_media NUMERIC(10,2);  
ALTER TABLE Fato_Vendas ADD COLUMN dia INTEGER;  
ALTER TABLE Fato_Vendas ADD COLUMN dia_semana INTEGER;  
ALTER TABLE Fato_Vendas ADD COLUMN mes INTEGER;  
ALTER TABLE Fato_Vendas ADD COLUMN ano INTEGER;
```

```
ALTER TABLE Fato_Compras ADD COLUMN mes INTEGER  
ALTER TABLE Fato_Compras ADD COLUMN ano INTEGER;
```

```
CREATE VIEW FATO_ULTIMACOMPRA AS  
SELECT CPF_CLIENTE, NOME, DATEDIFF(CURDATE(), ULTIMA_VENDA) 'Dia ULTIMA VENDA'  
FROM DW.CLIENTES;
```

```
CREATE VIEW FATO_LUCRO AS  
SELECT PROD_COMPRA2 ID_PROD, MES_COMPRA2 ANO_MES,  
ANO_COMPRA2 ANO_NUMERO, DATAS2, IFNULL(COMPCOMP, 0) CUSTO,  
IFNULL(VENDVEND, 0) RECEITA FROM  
(SELECT COMP.PROD_COMPRA PROD_COMPRA2, COMP.MES_COMPRA  
MES_COMPRA2, COMP.ANO_COMPRA ANO_COMPRA2, COMP.DATAS2  
,SUM(COMP.COMPRAS) COMPCOMP FROM  
(SELECT C.TB012_COD_PRODUTO PROD_COMPRA, C.TB012_017_DATA DATAS2,  
MONTH(C.TB012_017_DATA) MES_COMPRA, YEAR(C.TB012_017_DATA) ANO_COMPRA,  
(SUM(C.TB012_017_QUANTIDADE)*SUM(C.TB012_017_VALOR_UNITARIO)) COMPRAS  
FROM DATAWAREHOUSE.TB012_017_COMPRAS C  
GROUP BY C.TB012_COD_PRODUTO, MONTH(C.TB012_017_DATA),  
YEAR(C.TB012_017_DATA), C.TB012_017_VALOR_UNITARIO) COMP  
GROUP BY COMP.PROD_COMPRA, COMP.MES_COMPRA,  
COMP.ANO_COMPRA) CMP LEFT JOIN
```

```

(SELECT  VEND.PROD_VENDA  PROD_VENDA2,  VEND.MES_VENDA
MES_VENDA2,  VEND.ANO_VENDA  ANO_VENDA2,  SUM(VEND.VENDAS)
VENDVEND FROM
(SELECT  V.TB012_COD_PRODUTO  PROD_VENDA,  MONTH(V.TB010_012_DATA)
MES_VENDA,  YEAR(V.TB010_012_DATA)  ANO_VENDA,
(SUM(V.TB010_012_VALOR_UNITARIO)*SUM(V.TB010_012_QUANTIDADE)) VENDAS
FROM DATAWAREHOUSE.TB010_012_VENDAS V
GROUP  BY  V.TB012_COD_PRODUTO,  MONTH(V.TB010_012_DATA),
YEAR(V.TB010_012_DATA), V.TB010_012_VALOR_UNITARIO) VEND
GROUP BY VEND.PROD_VENDA, VEND.MES_VENDA, VEND.ANO_VENDA) VND
ON  VND.PROD_VENDA2 =  CMP.PROD_COMPRA2 AND  VND.ANO_VENDA2 =
CMP.ANO_COMPRA2 AND VND.MES_VENDA2 = CMP.MES_COMPRA2;

```

```

INSERT INTO DW.CATEGORIA (ID_CATEGORIA, NOME)
SELECT TB013_COD_CATEGORIA, TB013_DESCRICAO
FROM DATAWAREHOUSE.TB013_CATEGORIAS;

```

```

INSERT INTO DW.PRODUTO (ID_PRODUTO, NOME, ID_CATEGORIA)
SELECT TB012_COD_PRODUTO, TB012_DESCRICAO, TB013_COD_CATEGORIA
FROM DATAWAREHOUSE.TB012_PRODUTOS;

```

```

INSERT INTO DW.CLIENTES (CPF_CLIENTE, NOME, ULTIMA_VENDA)
SELECT TB010_CPF, TB010_NOME, NULL
FROM DATAWAREHOUSE.TB010_CLIENTES;

```

```

INSERT INTO DW.DATA (ID_DATA, DATA, DIA, MES, ANO)
SELECT  TB010_012_COD_VENDA,  TB010_012_DATA,  DAY(TB010_012_DATA),
MONTH(TB010_012_DATA), YEAR(TB010_012_DATA)
FROM DATAWAREHOUSE.TB010_012_VENDAS;

```

```

INSERT INTO DW.FORNECEDOR (ID_FORNECEDOR, NOME)
SELECT TB017_COD_FORNECEDOR, TB017_NOME_FANTASIA
FROM DATAWAREHOUSE.TB017_FORNECEDORES;

```



```

INSERT INTO DW.FUNCIONARIO (MATRICULA, NOME)
SELECT TB005_MATRICULA, TB005_NOME_COMPLETO
FROM DATAWAREHOUSE.TB005_FUNCIONARIOS;

```

```

UPDATE DW.CLIENTES SET ULTIMA_VENDA =
(SELECT MAX(V.TB010_012_DATA)
FROM DATAWAREHOUSE.TB010_012_VENDAS V
WHERE V.TB010_CPF = CPF_CLIENTE
GROUP BY V.TB010_CPF);

```

```

INSERT INTO DW.FATO_COMPRAS (QUANTIDADE, VALOR, ID_PRODUTO, ID_DATA,
ID_FORNECEDOR, MES, ANO)
SELECT SUM(C.TB012_017_QUANTIDADE), SUM(C.TB012_017_VALOR_UNITARIO),
C.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA, C.TB017_COD_FORNECEDOR,
MONTH(C.TB012_017_DATA), YEAR(C.TB012_017_DATA)
FROM DATAWAREHOUSE.TB012_017_COMPRAS C, DATAWAREHOUSE.TB010_012_VENDAS V
GROUP BY C.TB012_COD_PRODUTO;

```

```

INSERT INTO DW.FATO_COMPRAS (QUANTIDADE, VALOR, ID_PRODUTO, ID_DATA,
ID_FORNECEDOR, MES, ANO)
SELECT SUM(C.TB012_017_QUANTIDADE), SUM(C.TB012_017_VALOR_UNITARIO),
C.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA, C.TB017_COD_FORNECEDOR, NULL,
YEAR(C.TB012_017_DATA)
FROM DATAWAREHOUSE.TB012_017_COMPRAS C, DATAWAREHOUSE.TB010_012_VENDAS V
GROUP BY C.TB012_COD_PRODUTO, YEAR(C.TB012_017_DATA);

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO)
SELECT SUM(V.TB010_012_QUANTIDADE), SUM(V.TB010_012_VALOR_UNITARIO),
V.TB010_CPF, V.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA, P.TB013_COD_CATEGORIA,
NULL, DAY(V.TB010_012_DATA), WEEKDAY(V.TB010_012_DATA), MONTH(V.TB010_012_DATA),
YEAR(V.TB010_012_DATA)
FROM DATAWAREHOUSE.TB010_012_VENDAS V, DATAWAREHOUSE.TB012_PRODUTOS P,
DATAWAREHOUSE.TB005_FUNCIONARIOS F, DATAWAREHOUSE.TB004_LOJAS L

```

```

WHERE P.TB012_COD_PRODUTO = V.TB012_COD_PRODUTO AND V.TB005_MATRICULA =
F.TB005_MATRICULA AND F.TB004_COD_LOJA = L.TB004_COD_LOJA
GROUP BY V.TB012_COD_PRODUTO, V.TB010_012_DATA, L.TB004_COD_LOJA;

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO)
SELECT      SUM(V.TB010_012_QUANTIDADE),      SUM(V.TB010_012_VALOR_UNITARIO),
V.TB010_CPF, V.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA, P.TB013_COD_CATEGORIA,
NULL,      NULL,      WEEKDAY(V.TB010_012_DATA),      MONTH(V.TB010_012_DATA),
YEAR(V.TB010_012_DATA)
FROM    DATAWAREHOUSE.TB010_012_VENDAS V, DATAWAREHOUSE.TB012_PRODUTOS P,
DATAWAREHOUSE.TB005_FUNCIONARIOS F, DATAWAREHOUSE.TB004_LOJAS L
WHERE P.TB012_COD_PRODUTO = V.TB012_COD_PRODUTO AND V.TB005_MATRICULA =
F.TB005_MATRICULA AND F.TB004_COD_LOJA = L.TB004_COD_LOJA
GROUP      BY      V.TB012_COD_PRODUTO,      WEEKDAY(V.TB010_012_DATA),
MONTH(V.TB010_012_DATA), YEAR(V.TB010_012_DATA), L.TB004_COD_LOJA;

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO)
SELECT      SUM(V.TB010_012_QUANTIDADE),      SUM(V.TB010_012_VALOR_UNITARIO),
V.TB010_CPF, V.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA, P.TB013_COD_CATEGORIA,
NULL, NULL, NULL, MONTH(V.TB010_012_DATA), YEAR(V.TB010_012_DATA)
FROM    DATAWAREHOUSE.TB010_012_VENDAS V, DATAWAREHOUSE.TB012_PRODUTOS P,
DATAWAREHOUSE.TB005_FUNCIONARIOS F, DATAWAREHOUSE.TB004_LOJAS L
WHERE P.TB012_COD_PRODUTO = V.TB012_COD_PRODUTO AND V.TB005_MATRICULA =
F.TB005_MATRICULA AND F.TB004_COD_LOJA = L.TB004_COD_LOJA
GROUP      BY      V.TB012_COD_PRODUTO,      MONTH(V.TB010_012_DATA),
YEAR(V.TB010_012_DATA), L.TB004_COD_LOJA;

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO)
SELECT      SUM(V.TB010_012_QUANTIDADE),      SUM(V.TB010_012_VALOR_UNITARIO),
V.TB010_CPF, V.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA, P.TB013_COD_CATEGORIA,
NULL, NULL, NULL, NULL, YEAR(V.TB010_012_DATA)

```

```

FROM DATAWAREHOUSE.TB010_012_VENDAS V, DATAWAREHOUSE.TB012_PRODUTOS P,
DATAWAREHOUSE.TB005_FUNCIONARIOS F, DATAWAREHOUSE.TB004_LOJAS L
WHERE P.TB012_COD_PRODUTO = V.TB012_COD_PRODUTO AND V.TB005_MATRICULA =
F.TB005_MATRICULA AND F.TB004_COD_LOJA = L.TB004_COD_LOJA
GROUP BY V.TB012_COD_PRODUTO, YEAR(V.TB010_012_DATA), L.TB004_COD_LOJA;

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO)
SELECT SUM(V.TB010_012_QUANTIDADE) QUANTIDADE, SUM(V.TB010_012_VALOR_UNITARIO)
VALOR, V.TB010_CPF, V.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA,
P.TB013_COD_CATEGORIA,
(V.TB010_012_VALOR_UNITARIO/DAYOFMONTH(LAST_DAY(V.TB010_012_DATA))) AS
MEDIA_POR_DIA, NULL, NULL, MONTH(V.TB010_012_DATA), YEAR(V.TB010_012_DATA)
FROM DATAWAREHOUSE.TB010_012_VENDAS V, DATAWAREHOUSE.TB012_PRODUTOS P,
DATAWAREHOUSE.TB005_FUNCIONARIOS F, DATAWAREHOUSE.TB004_LOJAS L
WHERE P.TB012_COD_PRODUTO = V.TB012_COD_PRODUTO AND V.TB005_MATRICULA =
F.TB005_MATRICULA AND F.TB004_COD_LOJA = L.TB004_COD_LOJA
GROUP BY V.TB010_CPF, MONTH(V.TB010_012_DATA)
ORDER BY V.TB010_012_VALOR_UNITARIO DESC LIMIT 30;

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO)
SELECT SUM(V.TB010_012_QUANTIDADE) QUANTIDADE, SUM(V.TB010_012_VALOR_UNITARIO)
VALOR, V.TB010_CPF, V.TB012_COD_PRODUTO, V.TB010_012_COD_VENDA,
P.TB013_COD_CATEGORIA, (V.TB010_012_VALOR_UNITARIO/12) AS MEDIA_POR_MES, NULL, NULL,
NULL, YEAR(V.TB010_012_DATA)
FROM DATAWAREHOUSE.TB010_012_VENDAS V, DATAWAREHOUSE.TB012_PRODUTOS P,
DATAWAREHOUSE.TB005_FUNCIONARIOS F, DATAWAREHOUSE.TB004_LOJAS L
WHERE P.TB012_COD_PRODUTO = V.TB012_COD_PRODUTO AND V.TB005_MATRICULA =
F.TB005_MATRICULA AND F.TB004_COD_LOJA = L.TB004_COD_LOJA
GROUP BY V.TB010_CPF, YEAR(V.TB010_012_DATA)
ORDER BY V.TB010_012_VALOR_UNITARIO DESC LIMIT 30;

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO, MATRICULA)
SELECT COUNT(TB010_012_DATA) QUANTIDADE, NULL, TB010_CPF, TB012_COD_PRODUTO,
TB010_012_COD_VENDA, NULL, NULL, DAY(TB010_012_DATA) DIA, WEEKDAY(TB010_012_DATA),
MONTH(TB010_012_DATA) MES, YEAR(TB010_012_DATA) ANO, TB005_MATRICULA MATRICULA
FROM DATAWAREHOUSE.TB010_012_VENDAS
GROUP BY TB005_MATRICULA, YEAR(TB010_012_DATA), MONTH(TB010_012_DATA),
DAY(TB010_012_DATA);

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO, MATRICULA)
SELECT COUNT(TB010_012_DATA) QUANTIDADE, NULL, TB010_CPF, TB012_COD_PRODUTO,
TB010_012_COD_VENDA, NULL, NULL, NULL, NULL, MONTH(TB010_012_DATA) MES,
YEAR(TB010_012_DATA) ANO, TB005_MATRICULA MATRICULA
FROM DATAWAREHOUSE.TB010_012_VENDAS
GROUP BY TB005_MATRICULA, YEAR(TB010_012_DATA), MONTH(TB010_012_DATA);

```

```

INSERT INTO DW.FATO_VENDAS (QUANTIDADE, VALOR, CPF_CLIENTE, ID_PRODUTO, ID_DATA,
ID_CATEGORIA, GASTO_MEDIA, DIA, DIA_SEMANA, MES, ANO, MATRICULA)
SELECT COUNT(TB010_012_DATA) QUANTIDADE, NULL, TB010_CPF, TB012_COD_PRODUTO,
TB010_012_COD_VENDA, NULL, NULL, NULL, NULL, NULL, YEAR(TB010_012_DATA) ANO,
TB005_MATRICULA MATRICULA
FROM DATAWAREHOUSE.TB010_012_VENDAS
GROUP BY TB005_MATRICULA, YEAR(TB010_012_DATA);

```

```

INSERT INTO DW.FATO_LUCRATIVIDADE (ANO, MES, ID_DATA, ID_PRODUTO, LUCRO)
SELECT F.ANO_NUMERO, F.ANO_MES, V.TB010_012_COD_VENDA, F.ID_PROD,
(F.RECEITA - F.CUSTO)
FROM FATO_LUCRO F, DATAWAREHOUSE.TB010_012_VENDAS V;

```

```

INSERT INTO DW.FATO_LUCRATIVIDADE (ANO, MES, ID_DATA, ID_PRODUTO, LUCRO)
SELECT F.ANO_NUMERO, NULL, V.TB010_012_COD_VENDA, F.ID_PROD,
(SUM(F.RECEITA)-SUM(F.CUSTO)) 'LUCRO ANUAL'
FROM FATO_LUCRO F, DATAWAREHOUSE.TB010_012_VENDAS V

```

```
GROUP BY F.ID_PROD, F.ANO_NUMERO;
```