



Base de Dados II

2014/2015

Data Warehouse

Realizado por:

1010832 – João Coelho

1010985 – Sandro Marques

1010834 – Vasco Fortuna

Sumário

Definição do produto e plano geral do projeto	4
Visão do produto	4
Relevância e interesse	4
Plano geral do projeto	4
Esquema do Data Warehouse	5
Estrela das vendas	5
Estrela das compras	6
Estrela das promoções	6
Base de Dados operacional	7
Ficheiro externos	8
Carregamento de ficheiros externos	8
CHANNELS.csv	8
employees.csv	8
Dicionário de dados	9
Fac_Vendas	9
Fac_Compras	9
Fac_Promocoes	10
Cálculo de registos	11
Dim_canais	11
Dim_clientes	11
Dim_empregados	11
Dim_fornecedores	11
Dim_Produtos	11
Dim_Promocoes	11
Dim_Tempo	12
Mini_Dim_Clientes	12
Fac_Vendas	12
Fac_Compras	12
Fac_Promocoes	12
Cálculo total:	12
Workbooks	13
Mostrar o lucro em função do mês em cada canal	13
Devolve o máximo e mínimo do custo por fornecedor (com a possibilidade de <i>drill down</i> e <i>roll up</i>)	13
Devolve os lucros em relação ao tempo (detalhado) com opção de <i>roll up</i>	14

Devolve as Categorias mais vendidas num certo mês, com possibilidade de <i>roll up</i> e <i>drill down</i>	14
Bibliografia	16
Anexos	17
Computador A (João)	17
Sequências.....	17
Tabelas	17
Permissões	17
Index's	18
Constraints	18
Constraints de referência	18
Computador B (Sandro)	19
Criação da ligação à base de dados operacional.....	19
Sequências.....	19
Tabelas	19
Index's	20
Constraints	20
Constraints de referência	22
Triggers.....	22
Funções	22
Procedimentos	25
Sinónimos	36
Computador C (Vasco)	37
Criação da ligação à base de dados operacional.....	37
Sequências.....	37
Tabelas	37
Index's	39
Constraints	39
Constraints de referência	40
Permissões	41
Triggers.....	41
Registos complementares	42

Definição do produto e plano geral do projeto

Visão do produto

O produto será constituído por uma data warehouse, que tem como objectivo, facilitar a pesquisa de registos e dados relevantes de uma base de dados. A data warehouse tratará e actualizará os registos da base de dados todos os dias (no período em que a base de dados tenha menor utilização), de modo a providenciar pesquisas e dados constantemente atualizados.

Software necessário: Oracle Business Intelligence Discoverer

Hardware recomendado:

- Memória: 4GB
- Disco rígido: 1TB
- CPU: dual-core Pentium, 1.5GHz

Relevância e interesse

A nível empresarial, a data warehouse traz grandes vantagens no tratamento de dados para estatísticas e para gestão de negócios em relação à base de dados que está a ser tratada. Adicionalmente, favorece os relatórios, a análise de grandes volumes de dados e a obtenção de informações estratégicas que podem facilitar a tomada de decisão.

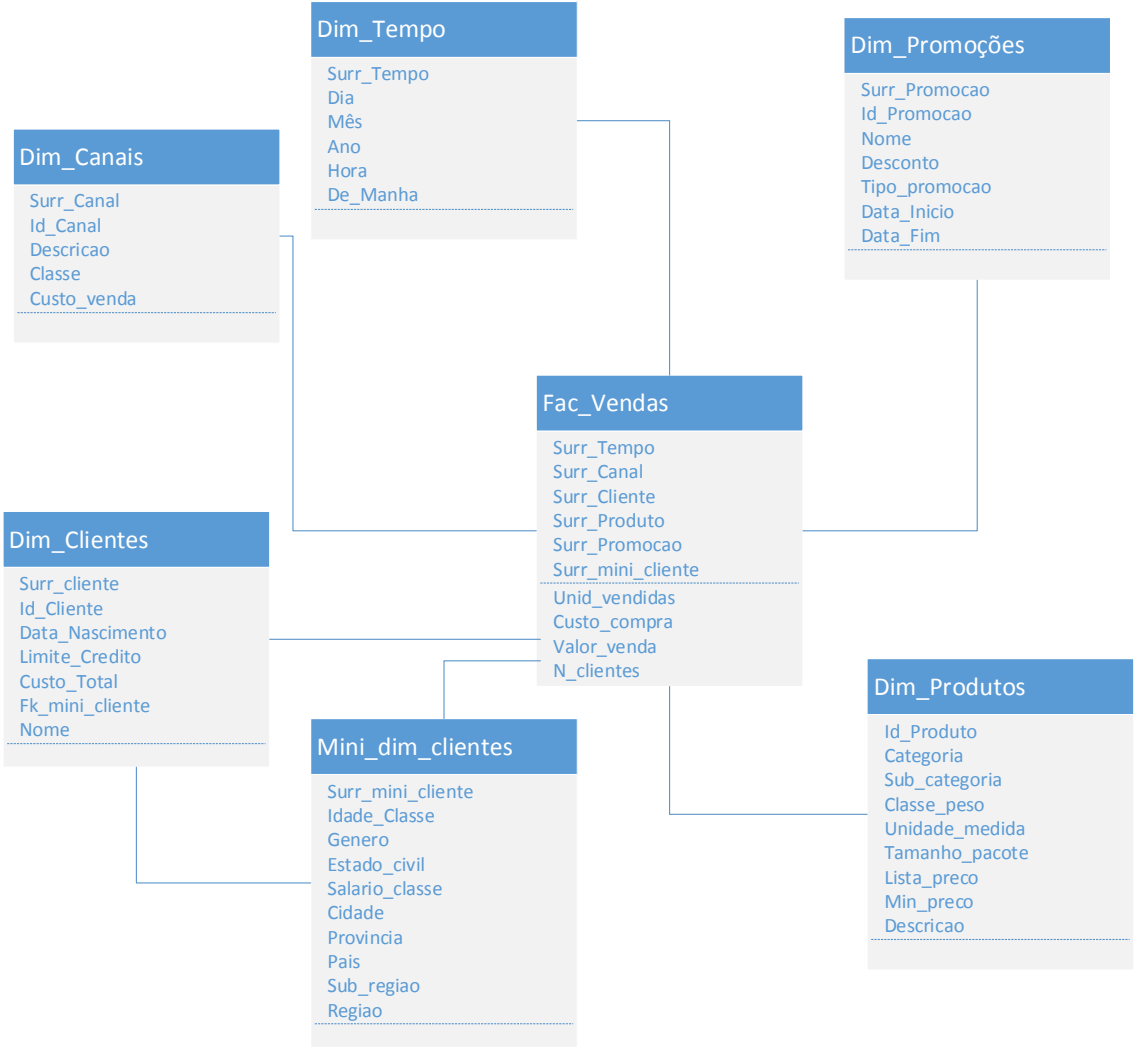
Plano geral do projeto

<i>ID</i>	<i>Nome da tarefa</i>	<i>Início</i>	<i>Conclusão</i>	<i>Duração</i>
1	Definição do produto e plano geral	26-03-2015	04-04-2015	1h
2	Esquema do data warehouse	26-03-2015	04-04-2015	2h
3	Descrição e código da componente ETL	04-04-2015	06-05-2015	40h
4	EUL e	04-05-2015	07-05-2015	10h

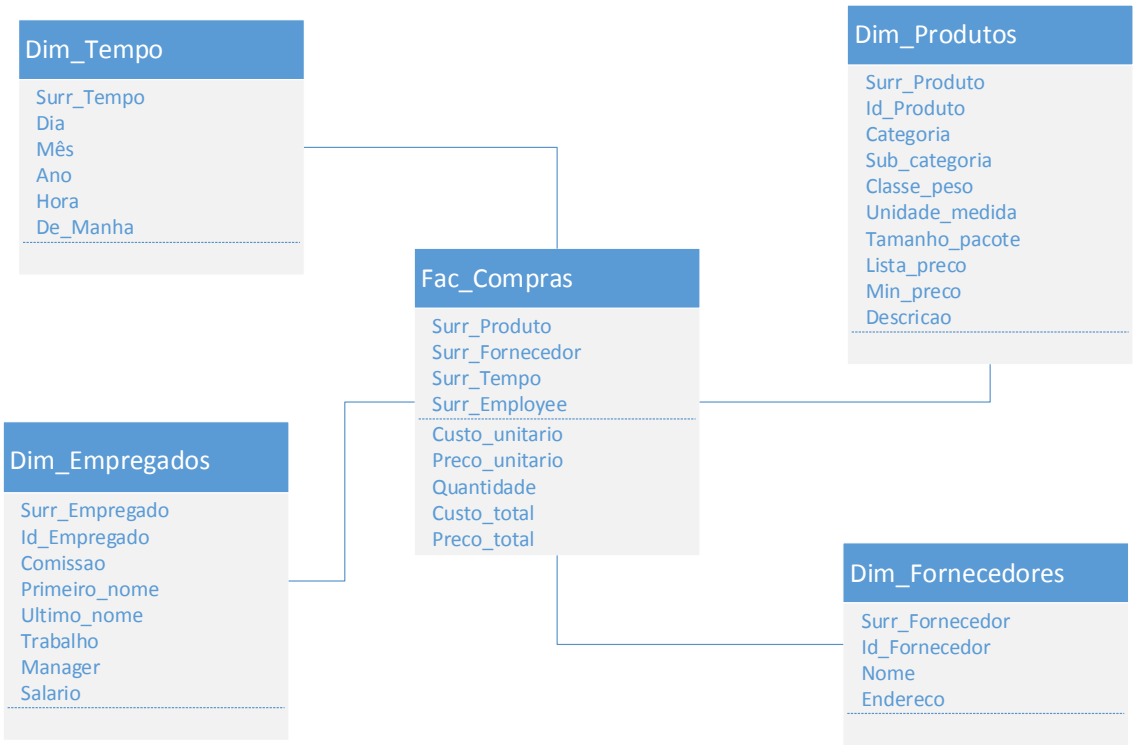
	workbooks no Discoverer			
5	Documento de otimização de desempenho			

Esquema do Data Warehouse

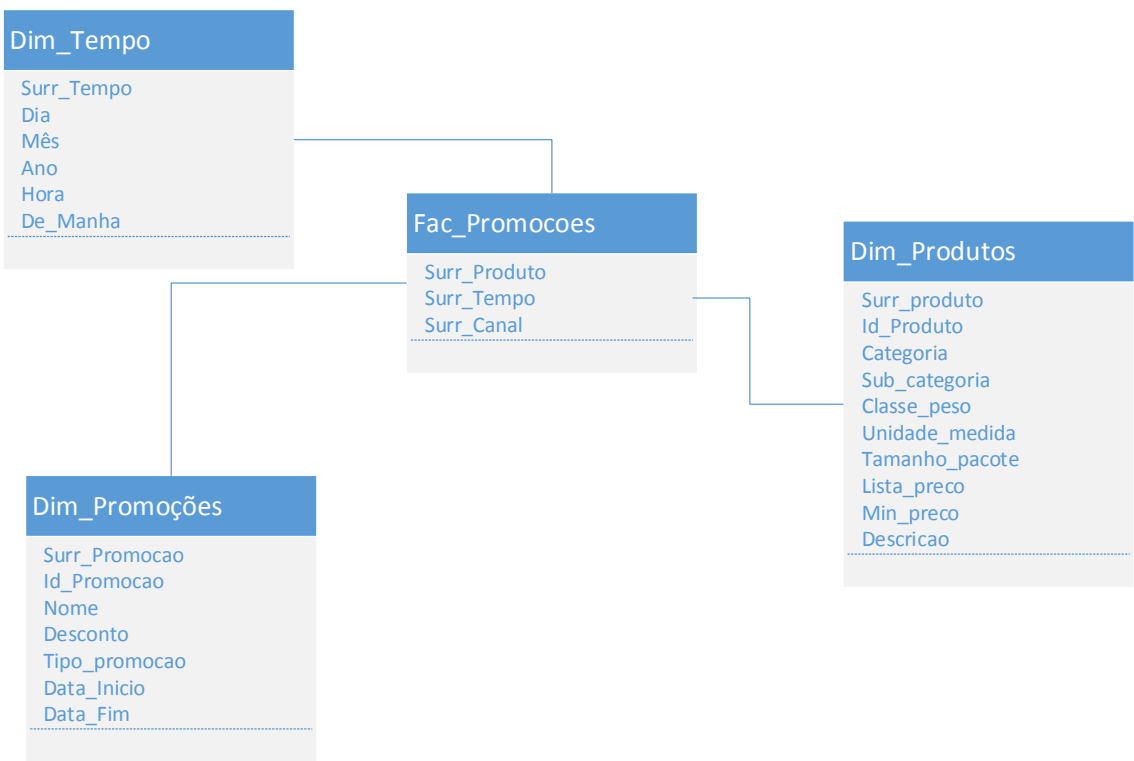
Estrela das vendas



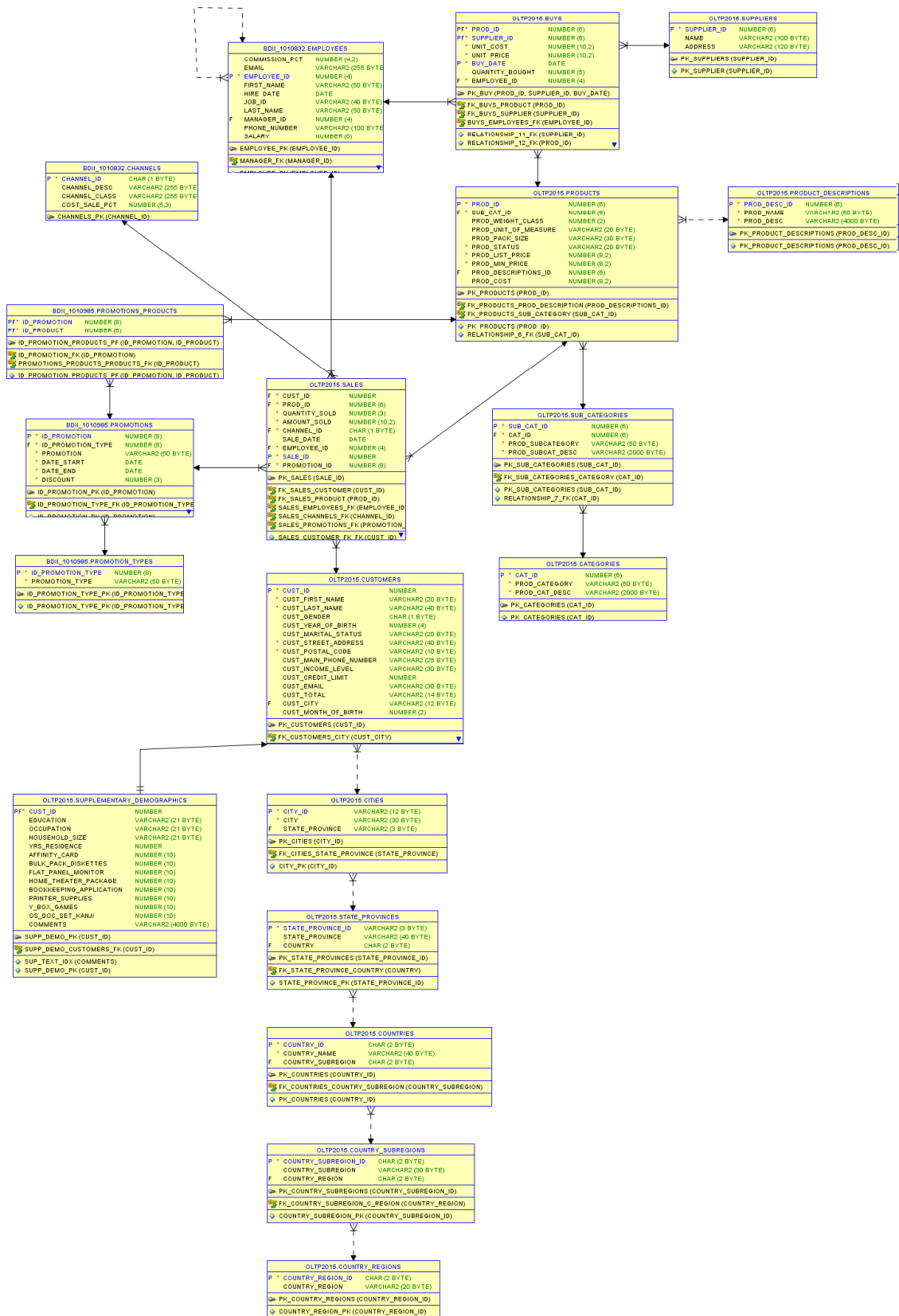
Estrela das compras



Estrela das promoções



Base de Datos operacional



Ficheiro externos

Carregamento de ficheiros externos

CHANNELS.csv

```
OPTIONS (SKIP=1)
LOAD DATA
INFILE 'CHANNELS.csv'
REPLACE
INTO TABLE bdii_1010832.CHANNELS
FIELDS TERMINATED BY ';' OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS
(CHANNEL_ID CHAR(1),
CHANNEL_DESC CHAR(255),
CHANNEL_CLASS CHAR(255),
COST_SALE_PCT decimal EXTERNAL)
```

employees.csv

```
OPTIONS (SKIP=1)
LOAD DATA
INFILE 'employees.csv'
REPLACE
INTO TABLE bdii_1010832.employees
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS
(COMMISSION_PCT decimal EXTERNAL,
EMAIL CHAR(50),
EMPLOYEE_ID integer external,
FIRST_NAME INTEGER EXTERNAL,
HIRE_DATE DATE "dd-MON-rr",
JOB_ID CHAR(40),
LAST_NAME CHAR(50),
MANAGER_ID integer external,
PHONE_NUMBER CHAR(100),
SALARY integer external
)
```


Dicionário de dados

Fac_Vendas

Tabela com as vendas de todos os canais

Nome	Tipo Dados	Tam.	Chave	Descrição
Surr_Tempo	Number		PF	Id do tempo
Surr_Canal	Number		PF	Id do canal
Surr_Cliente	Number		PF	Id do cliente
Surr_Promocao	Number		PF	Id da promoção
Surr_Produto	Number		PF	Id do produto
Surr_mini_cliente	Number		PF	Id da mini dimensão cliente
Surr_empregado	Number		PF	Id do empregado
Unid_Vendidas	Number			Nº de unidades vendidas
Custo_Compra	Number			Preço de custo total (custo_prod * quantity_sold)
Valor_Venda	Number			Valor total da venda
Lucro	Number			Lucro da venda (Valor_Venda – Custo_Compra)

Fac_Compras

Tabela com as compras aos fornecedores

Nome	Tipo Dados	Tam.	Chave	Descrição
Surr_Produto	Number		PF	Id do produto
Surr_Fornecedor	Number		PF	Id do canal
Surr_Tempo	Number		PF	Id do tempo
Surr_Employee	Number		PF	Id do cliente
Custo_Unitario	Number			Preço de custo por unidade
Preco_Unitario	Number			Preço de venda por unidade
Quantidade	Number			Quantidade adquirida

Custo_Total	Number			Custo total da compra (Custo_Unitario * Quantidade)
Preco_Total	Number			Preço total se escoar o produto (Preco_Unitario * Quantidade)

Fac_Promocoes

Tabela com as promoções de todos os canais

Nome	Tipo Dados	Tam	Chave	Descrição
Surr_Produto	Number		PF	Id do produto
Surr_Tempo	Number		PF	Id do tempo
Surr_Canal	Number		PF	Id do canal

Cálculo de registros

Dimensão máxima de um “number”: 21 bytes

Dim_canaís

Número de campos: 5

Número de registros estimados: 5 por ano

Total ocupado: $21 \text{ (bytes)} * 5 * 5 : 525 \text{ bytes}$

Dim_clientes

Número de campos: 7

Número de registros estimados: 50000 por ano

Total ocupado: $21 \text{ (bytes)} * 7 * 50000 : 7350000 \text{ bytes}$

Dim_empregados

Número de campos: 8

Número de registros estimados: 100 por ano

Total ocupado: $21 \text{ (bytes)} * 8 * 100 : 16800 \text{ bytes}$

Dim_fornecedores

Número de campos: 4

Número de registros estimados: 300 por ano

Total ocupado: $21 \text{ (bytes)} * 4 * 300 : 25200 \text{ bytes}$

Dim_Produtos

Número de campos: 12

Número de registros estimados: 10000 por ano

Total ocupado: $21 \text{ (bytes)} * 12 * 10000 : 2520000 \text{ bytes}$

Dim_Promocoes

Número de campos: 7

Número de registros estimados: 40 por ano

Total ocupado: $21 \text{ (bytes)} * 7 * 40 : 2520000 \text{ bytes}$

Dim_Tempo

Número de campos: 6

Número de registos estimados: 365 por ano

Total ocupado: 21 (bytes) * 6*365: 45990 bytes

Mini_Dim_Clientes

Número de campos: 10

Número de registos estimados: 25000 por ano

Total ocupado: 21 (bytes) * 10 * 25000: 5250000 bytes

Fac_Vendas

Número de campos: 11

Número de registos estimados: 1300000 por ano

Total ocupado: 21 (bytes) * 11 * 1300000: 300300000 bytes

Fac_Compras

Número de campos: 9

Número de registos estimados: 1000000 por ano

Total ocupado: 21 (bytes) * 11 * 1000000: 231000000 bytes

Fac_Promocoes

Número de campos: 3

Número de registos estimados: 9000000 por ano

Total ocupado: 21 (bytes) * 3 * 9000000: 567000000 bytes

Cálculo total:

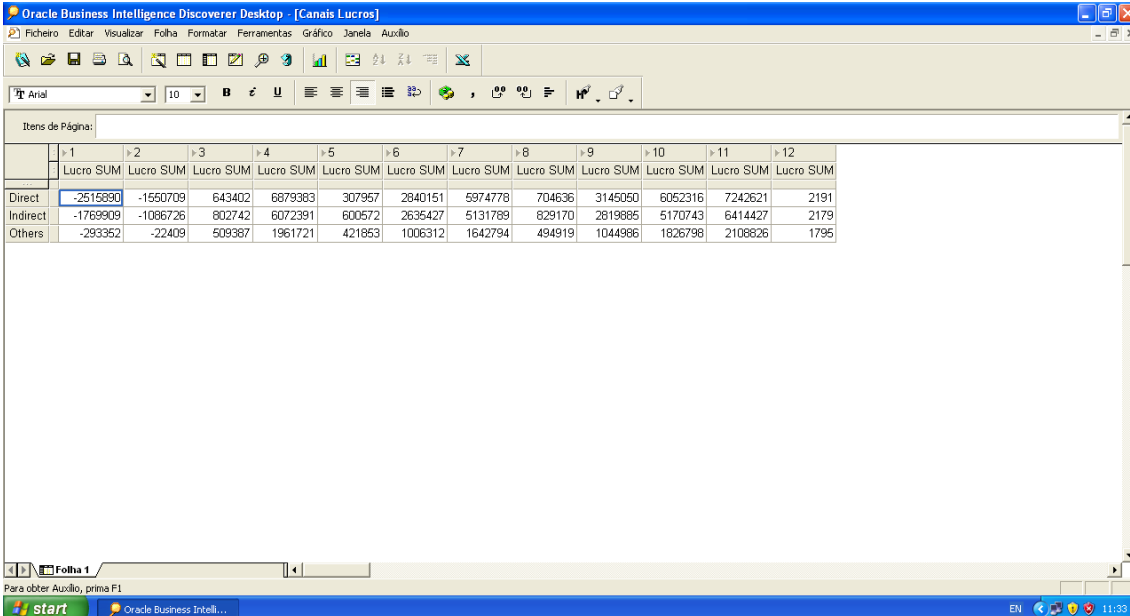
300300000+231000000+567000000: 3801000000bytes

Devido à grande dimensão dos factos, o espaço ocupado pelas dimensões foi ignorado, o resultado final aproxima-se aos **3,5GB** por ano, estimando ocupar cerca de **35GB** em 10 anos.

Workbooks

Mostrar o lucro em função do mês em cada canal

```
SELECT O100022.CLASSE, O100146.MES, SUM(O100170.LUCRO)
FROM BDII_1010834.DIM_CANAIS O100022, BDII_1010834.DIM_TEMPO O100146,
BDII_1010834.FAC_VENDAS O100170
WHERE ( ( O100022.SURR_CANAL = O100170.SURR_CANAL ) AND
( O100146.SURR_TEMPO = O100170.SURR_TEMPO ) ) AND ( O100146.ANO
= : "ano" )
GROUP BY O100022.CLASSE, O100146.MES
;
```



The screenshot shows the Oracle Business Intelligence Discoverer Desktop interface. The main window displays a pivot table titled 'Canais Lucros'. The table has 13 columns representing months from 1 to 12, and 4 rows representing different channels: Direct, Indirect, Others, and a blank row. The data is summarized by 'Lucro SUM'.

	>1	>2	>3	>4	>5	>6	>7	>8	>9	>10	>11	>12
Direct	-2515890	-1550709	643402	6879383	307957	2840151	5974778	704636	3145050	6052316	7242621	2191
Indirect	-1769909	-1086726	802742	6072391	600572	2635427	5131789	829170	2819885	5170743	6414427	2179
Others	-293352	-22409	509387	1961721	421853	1006312	1642794	494919	1044986	1826798	2108826	1795

Devolve o máximo e mínimo do custo por fornecedor (com a possibilidade de drill down e roll up)

```
SELECT O100071.NOME, O100077.CATEGORIA, O100146.ANO,
MAX(O100154.PRECO_UNITARIO), MIN(O100154.PRECO_UNITARIO)
FROM BDII_1010834.DIM_FORNECEDORES O100071, BDII_1010834.DIM_PRODUTOS
O100077, BDII_1010834.DIM_TEMPO O100146, BDII_1010834.FAC_COMPRAS
O100154
WHERE ( ( O100071.SURR_FORNECEDOR = O100154.SURR_FORNECEDOR ) AND
( O100077.SURR_PRODUTO = O100154.SURR_PRODUTO ) AND
( O100146.SURR_TEMPO = O100154.SURR_TEMPO ) )
GROUP BY O100071.NOME, O100077.CATEGORIA, O100146.ANO
;
```

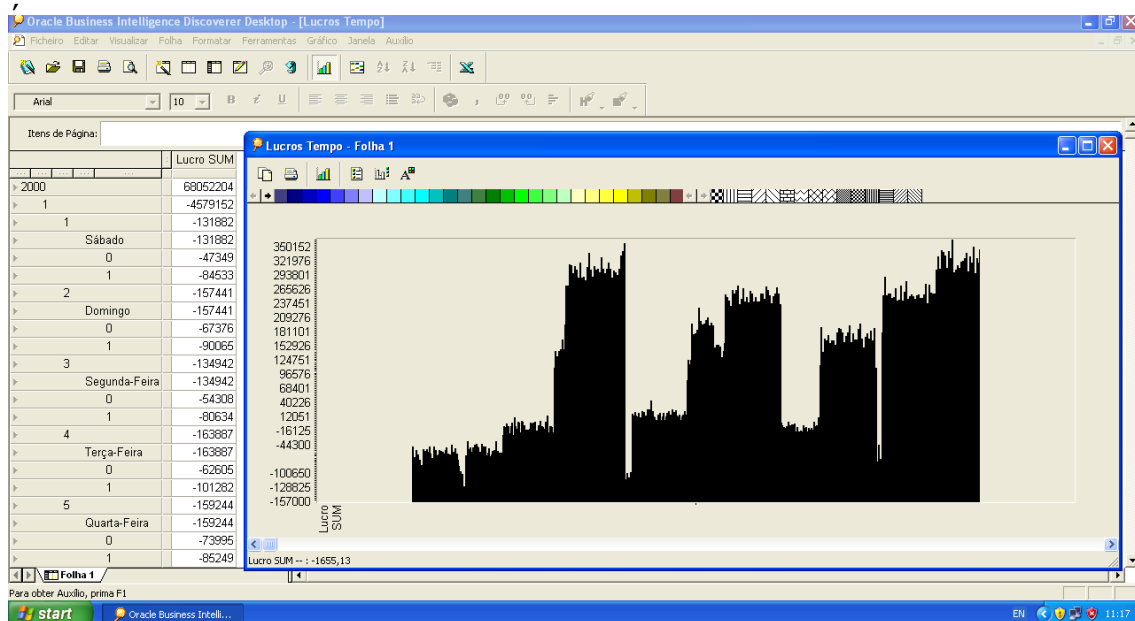
Oracle Business Intelligence Discoverer Desktop - [Fornecedores max min preco por produto]

Para obter Auxílio, prima F1

Itens de Página:	Boys		Girls		Men		Women	
	Preço Unitário MIN	Preço Unitário MAX	Preço Unitário MIN	Preço Unitário MAX	Preço Unitário MIN	Preço Unitário MAX	Preço Unitário MIN	Preço Unitário MAX
d9tLwUZYnQOZtl cTozgHpNrpYeTMjVB	2,10	405,74	4,00	253,57	2,10	491,09	4,04	298,48
qbWcObrdFw uOsJbv	2,10	255,18	9,88	375,04	5,39	455,12	5,44	658,19
BaJkpz aEJpoxhYNCKNBhO	2,36	816,57	8,47	1007,03	4,03	275,00	10,08	1104,50
wHfzZbDTR THyaGLteYOboBe	2,40	810,71	13,63	518,52	3,33	686,33	6,19	497,81
CIkbVmmcum pwh	2,63	490,28	7,60	334,60	2,82	557,11	6,12	265,95
MipGlZwCtLHeTaTzNC akA	2,72	549,70	10,50	501,88	7,90	470,51	4,22	810,96
CSxXORdeWSzkOMC Ssoajrl	3,00	859,81	4,80	266,54	7,33	854,56	5,39	397,54
izNgBVDHbualmDO EroW	3,00	223,73	6,37	1093,33	5,00	408,53	7,83	463,40
jShdAtOpbTEYAEUj XoCzX	3,00	378,27	6,69	589,51	15,25	936,35	5,10	630,81
NaxsaarkcteYKHx FdSDpNmLNdT	3,00	472,25	7,20	390,18	6,89	683,96	8,00	516,21
SJGCRMRtgYBkAG TdDabUHaa	3,00	341,38	2,10	394,04	4,08	906,36	3,00	363,63
tThDHQwXLaZShpP plwfuXYD	3,00	848,18	4,07	654,97	5,25	539,28	5,01	269,84
LkQDEeDlZgECMAm VsCnkogbdhpXziN	3,02	420,59	12,11	820,57	2,71	416,98	3,84	434,69
ZcYumTutpSHSe UFksoplBgh	3,10	377,86	4,66	264,15	3,89	483,29	4,80	686,16
ikHVPjppTWZ SXORjSkUomIR	3,20	339,46	8,11	799,09	8,03	411,31	9,86	497,87
XlKLnkmiHUWmBP zEHjPbaEDgmYU	3,20	1037,49	8,00	360,41	7,72	407,24	6,64	387,27
pTYZeNZoNN zS	3,28	476,09	5,99	453,48	6,40	706,72	15,47	562,81
AmLNOGaHeSeTaXd wLBhoGtvkNdaUsP	3,36	418,96	7,00	384,85	10,47	1103,96	8,10	994,82
pyPgScdtJhi FsMkQNmNAXb	3,38	233,15	7,05	373,67	4,11	432,82	8,24	537,92
sOnayHCmOqWMZxO NHmxd	3,53	398,68	5,25	405,32	14,92	372,37	5,82	2551,52
rD mnhyDBGMcK	3,55	708,18	7,66	662,96	3,12	368,79	8,18	652,15

Devolve os lucros em relação ao tempo (detalhado) com opção de roll up

```
SELECT O100071.NOME, O100077.CATEGORIA, O100146.ANO,
MAX(O100154.PRECO_UNITARIO), MIN(O100154.PRECO_UNITARIO)
FROM BDII_1010834.DIM_FORNECEDORES O100071, BDII_1010834.DIM_PRODUTOS
O100077, BDII_1010834.DIM_TEMPO O100146, BDII_1010834.FAC_COMPRAS
O100154
WHERE ( ( O100071.SURR_FORNECEDOR = O100154.SURR_FORNECEDOR ) AND
( O100077.SURR_PRODUTO = O100154.SURR_PRODUTO ) AND
( O100146.SURR_TEMPO = O100154.SURR_TEMPO ) )
GROUP BY O100071.NOME, O100077.CATEGORIA, O100146.ANO
```



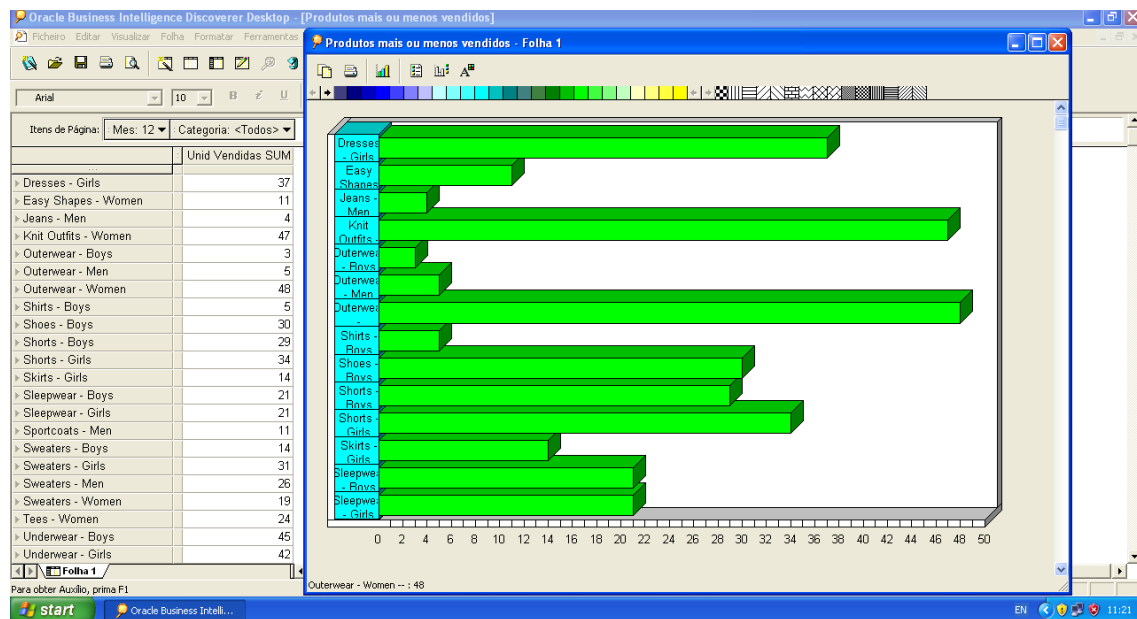
Devolve as Categorias mais vendidas num certo mês, com possibilidade de roll up e drill down

```
SELECT O100077.CATEGORIA, O100077.SUB_CATEGORIA, O100146.MES,
SUM(O100170.UNID_VENDIDAS)
```

```

FROM BDII_1010834.DIM_PRODUTOS O100077, BDII_1010834.DIM_TEMPO O100146,
BDII_1010834.FAC_VENDAS O100170
WHERE ( ( O100077.SURR_PRODUTO = O100170.SURR_PRODUTO ) AND
( O100146.SURR_TEMPO = O100170.SURR_TEMPO ) ) AND ( O100146.ANO
= : "ano" )
GROUP BY O100077.CATEGORIA, O100077.SUB_CATEGORIA, O100146.MES
HAVING ( ( SUM(O100170.UNID_VENDIDAS) ) < : "unidades vendidas
inferiores a" )
;

```



Bibliografia

OracleFAQs. (2015). *Number* [Em linha]. Disponível em <http://www.orafaq.com/wiki/Number>. [Consultado em 07-05-2015]

Tech on the Net. (2015). *Exception Handling* [Em linha]. Disponível em <http://www.techonthenet.com/oracle/exceptions/sqlerrm.php>. [Consultado em 07-05-2015]

Anexos

Computador A (João)

Sequências

```
CREATE SEQUENCE "BDII_1010832"."SEQ_VENDAS" MINVALUE 1 MAXVALUE  
999999999999999999999999999999999INCREMENT BY 1 START WITH 120 CACHE 20  
NOORDER NOCYCLE
```

Tabelas

```
CREATE TABLE "BDII_1010832"."CHANNELS" ("CHANNEL_ID" CHAR(1 BYTE) ,
"CHANNEL_DESC" VARCHAR2(255 BYTE) , "CHANNEL_CLASS" VARCHAR2(255 BYTE) ,
"COST_SALE_PCT" NUMBER(5,3)) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS
255 NOCOMPRESS LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1
MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER POOL DEFAULT) TABLESPACE "USERS"
```

```
CREATE TABLE "BDII_1010832"."EMPLOYEES" ("COMMISSION_PCT" NUMBER(4,2)
DEFAULT 0, "EMAIL" VARCHAR2(255 BYTE), "EMPLOYEE_ID" NUMBER(4,0),
"FIRST_NAME" VARCHAR2(50 BYTE), "HIRE_DATE" DATE, "JOB_ID" VARCHAR2(40
BYTE), "LAST_NAME" VARCHAR2(50 BYTE), "MANAGER_ID" NUMBER(4,0),
"PHONE_NUMBER" VARCHAR2(100 BYTE), "SALARY" NUMBER(6,0)) PCTFREE 10
PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING STORAGE(INITIAL
65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0
FREELISTS 1 FREELIST GROUPS 1 BUFFER POOL DEFAULT) TABLESPACE "USERS"
```

Permissões

```
GRANT DELETE ON "BDII 1010832"."CHANNELS" TO "BDII 1010834"
```

```
GRANT INSERT ON "BDII 1010832"."CHANNELS" TO "BDII 1010834"
```

```
GRANT SELECT ON "BDII 1010832"."CHANNELS" TO "BDII 1010834"
```

GRANT UPDATE ON "BDII 1010832"."CHANNELS" TO "BDII 1010834"

```
GRANT DELETE ON "BDII 1010832"."CHANNELS" TO "BDII 1010985"
```

```
GRANT INSERT ON "BDII 1010832"."CHANNELS" TO "BDII 1010985"
```

```
GRANT SELECT ON "BDII 1010832"."CHANNELS" TO "BDII 1010985"
```

GRANT UPDATE ON "BDII 1010832"."CHANNELS" TO "BDII 1010985"

```
GRANT DELETE ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010834"
```

```
GRANT INSERT ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010834"
```

```
GRANT SELECT ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010834"
```

GRANT UPDATE ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010834"

```
GRANT DELETE ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010985"
```

```
GRANT INSERT ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010985"
```

```
GRANT SELECT ON "BDII 1010832"."EMPLOYEES" TO "BDII 1010985"
```

```
GRANT UPDATE ON "BDII_1010832"."EMPLOYEES" TO "BDII_1010985"
```

Index's

```
CREATE UNIQUE INDEX "BDII_1010832"."EMPLOYEE_PK" ON  
"BDII_1010832"."EMPLOYEES" ("EMPLOYEE_ID") PCTFREE 10 INITRANS 2  
MAXTRANS 255 COMPUTE STATISTICS STORAGE (INITIAL 65536 NEXT 1048576  
MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST  
GROUPS 1 BUFFER_POOL DEFAULT) TABLESPACE "USERS"
```

```
CREATE UNIQUE INDEX "BDII_1010832"."PK_CHANNEL_ID" ON  
"BDII_1010832"."CHANNELS" ("CHANNEL_ID") PCTFREE 10 INITRANS 2  
MAXTRANS 255 COMPUTE STATISTICS STORAGE (INITIAL 65536 NEXT 1048576  
MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST  
GROUPS 1 BUFFER_POOL DEFAULT) TABLESPACE "USERS"
```

Constraints

```
ALTER TABLE "BDII_1010832"."EMPLOYEES" ADD CONSTRAINT "EMPLOYEE_PK"  
PRIMARY KEY ("EMPLOYEE_ID") USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS  
255 COMPUTE STATISTICS STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS  
1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT) TABLESPACE "USERS" ENABLE
```

```
ALTER TABLE "BDII_1010832"."CHANNELS" ADD CONSTRAINT "PK_CHANNEL_ID"  
PRIMARY KEY ("CHANNEL_ID") USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS  
255 COMPUTE STATISTICS STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS  
1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT) TABLESPACE "USERS" ENABLE
```

```
ALTER TABLE "BDII_1010832"."CHANNELS" MODIFY ("CHANNEL_ID" NOT NULL  
ENABLE)
```

Constraints de referência

```
ALTER TABLE "BDII_1010832"."EMPLOYEES" ADD CONSTRAINT "MANAGER_FK"  
FOREIGN KEY ("MANAGER_ID") REFERENCES "BDII_1010832"."EMPLOYEES"  
("EMPLOYEE_ID") ENABLE
```

Criação da ligação à base de dados operacional

Sequências

```
CREATE TABLE "BDII 1010985"."PROMOTION TYPES"
```

```

        (      "ID_PROMOTION_TYPE" NUMBER(8,0),
          "PROMOTION_TYPE" VARCHAR2(50 BYTE)
        ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ;

CREATE TABLE "BDII_1010985"."PROMOTIONS_PRODUCTS"
(      "ID_PROMOTION" NUMBER(8,0),
      "ID_PRODUCT" NUMBER(8,0)
) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ;

```

Index's

```

CREATE UNIQUE INDEX "BDII_1010985"."ID_PROMOTION_PRODUCTS_PF" ON
"BDII_1010985"."PROMOTIONS_PRODUCTS" ("ID_PROMOTION", "ID_PRODUCT")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ;

CREATE UNIQUE INDEX "BDII_1010985"."ID_PROMOTION_TYPE_PK" ON
"BDII_1010985"."PROMOTION_TYPES" ("ID_PROMOTION_TYPE")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ;

CREATE UNIQUE INDEX "BDII_1010985"."LOG_ERRORS_PK" ON
"BDII_1010985"."LOG_ERRORS" ("ERROR_ID_SK")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ;

CREATE UNIQUE INDEX "BDII_1010985"."ID_PROMOTION_PK" ON
"BDII_1010985"."PROMOTIONS" ("ID_PROMOTION")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ;

```

Constraints

```

ALTER TABLE "BDII_1010985"."PROMOTIONS_PRODUCTS" ADD CONSTRAINT
"ID_PROMOTION_PRODUCTS_PF" PRIMARY KEY ("ID_PROMOTION", "ID_PRODUCT")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645

```

```

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ENABLE;

ALTER TABLE "BDII_1010985"."PROMOTIONS_PRODUCTS" MODIFY
("ID_PROMOTION" NOT NULL ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS_PRODUCTS" MODIFY ("ID_PRODUCT"
NOT NULL ENABLE);

ALTER TABLE "BDII_1010985"."LOG_ERRORS" ADD CONSTRAINT
"LOG_ERRORS_PK" PRIMARY KEY ("ERROR_ID_SK")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ENABLE;

ALTER TABLE "BDII_1010985"."LOG_ERRORS" MODIFY ("ERROR_ID_SK" NOT
NULL ENABLE);

ALTER TABLE "BDII_1010985"."LOG_ERRORS" MODIFY ("ERROR_CODE" NOT
NULL ENABLE);

ALTER TABLE "BDII_1010985"."LOG_ERRORS" MODIFY ("ERROR_MSG" NOT NULL
ENABLE);

ALTER TABLE "BDII_1010985"."LOG_ERRORS" MODIFY ("OPERATION_NAME" NOT
NULL ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS" ADD CONSTRAINT
"ID_PROMOTION_PK" PRIMARY KEY ("ID_PROMOTION")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "USERS" ENABLE;

ALTER TABLE "BDII_1010985"."PROMOTIONS" MODIFY ("ID_PROMOTION" NOT
NULL ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS" MODIFY ("ID_PROMOTION_TYPE"
NOT NULL ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS" MODIFY ("PROMOTION" NOT NULL
ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS" MODIFY ("DATE_START" NOT
NULL ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS" MODIFY ("DATE_END" NOT NULL
ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTIONS" MODIFY ("DISCOUNT" NOT NULL
ENABLE);

ALTER TABLE "BDII_1010985"."PROMOTION_TYPES" ADD CONSTRAINT
"ID_PROMOTION_TYPE_PK" PRIMARY KEY ("ID_PROMOTION_TYPE")

```

```

        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
        STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
        PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
        TABLESPACE "USERS"    ENABLE;

    ALTER TABLE "BDII_1010985"."PROMOTION_TYPES" MODIFY
("ID_PROMOTION_TYPE" NOT NULL ENABLE);

    ALTER TABLE "BDII_1010985"."PROMOTION_TYPES" MODIFY
("PROMOTION_TYPE" NOT NULL ENABLE);

```

Constraints de referência

```

    ALTER TABLE "BDII_1010985"."PROMOTIONS" ADD CONSTRAINT
"ID_PROMOTION_TYPE_FK" FOREIGN KEY ("ID_PROMOTION_TYPE")
    REFERENCES "BDII_1010985"."PROMOTION_TYPES" ("ID_PROMOTION_TYPE")
    ENABLE;

    ALTER TABLE "BDII_1010985"."PROMOTIONS_PRODUCTS" ADD CONSTRAINT
"ID_PROMOTION_FK" FOREIGN KEY ("ID_PROMOTION")
    REFERENCES "BDII_1010985"."PROMOTIONS" ("ID_PROMOTION") ENABLE;

```

Triggers

```

    CREATE OR REPLACE TRIGGER "BDII_1010985"."ID_PROMOTIONS"
    before insert on promotions
    for each row
    Begin
        select seq_promotions.nextval
        into :new.id_promotion from dual;
    end;
/
    ALTER TRIGGER "BDII_1010985"."ID_PROMOTIONS" ENABLE;

    CREATE OR REPLACE TRIGGER "BDII_1010985"."ID_PROMOTION_TYPES"
    before insert on promotion_types
    for each row
    Begin
        select seq_promotion_type.nextval
        into :new.id_promotion_type from dual;
    end;
/
    ALTER TRIGGER "BDII_1010985"."ID_PROMOTION_TYPES" ENABLE;

    CREATE OR REPLACE TRIGGER "BDII_1010985"."TRIGGER_LOG_ERRORS"
    before insert on log_errors
    for each row
    begin
        select seq_log_errors.nextval into :new.error_id_sk from dual;
    end;
/
    ALTER TRIGGER "BDII_1010985"."TRIGGER_LOG_ERRORS" ENABLE;

```

Funções

```

--Esta função vai buscar é tabela idade_classe, a classe a que
--corresponde a idade passada por parâmetro

```

```

CREATE OR REPLACE FUNCTION "BDII_1010985"."CAL_IDADE_CLASSE" (year
number,month number) return varchar2 as
idade number;
classe varchar2(255);
idade_data date;
begin
    idade_data:= (last_day( to_date( to_char(year) || '-' ||
to_char(month),'yyyy-mm')));
    idade:= trunc(months_between(sysdate,idade_data)/12,0);
    select classe || ' : ' || minimo || ' - ' || maximo into classe
from BDII_1010834.classe_idade where minimo<=idade and idade<=maximo;
    return classe;
end;

/

```

```

--Função para criar um data aleatória entre '2015-01-01' e '2015-03-
16'

```

```

CREATE OR REPLACE FUNCTION "BDII_1010985"."GENERATE_RANDOM_DATE"
return date is
v_date date;--data a ser retornada
begin
    select to_date(
        trunc(
            --As duas datas são convertidas para julian e escolhe-se um valor
aleatorio entre as duas
            dbms_random.value(
                to_char(date '2015-01-01', 'J')
                ,to_char(date '2015-03-16', 'J')
            )
        ),'J'
    ) into v_date from dual;
    return v_date;
end;

/

```

```

--Devolve-nos a classe correspondente ao salário passado por
parâmetro
CREATE OR REPLACE FUNCTION "BDII_1010985"."GET_CLASSE_SALARIO"
(salario number) return varchar2 is
cursor cursor_classe_salario is
    select id_classe, classe from BDII_1010834.classe_salario;
var_classe varchar2(50); -- texto da classe (ex.: B: 30000 - 49999)
var_classe_temp varchar2(50);
var_valor_max number; -- valor máximo
var_id_classe number; -- id da classe
var_salario_anual number; -- salario * 12
begin
    var_salario_anual := salario*12;
    open cursor_classe_salario;
    loop
        fetch cursor_classe_salario into var_id_classe, var_classe;
        exit when cursor_classe_salario%NOTFOUND;
        var_classe_temp := replace(var_classe, ',', ' ');
        if (instr(var_classe_temp, 'Below') > 0) then
            -- é o 1º registo

```

```

        var_valor_max :=
to_number(substr(var_classe_temp, (instr(var_classe_temp,
'Below')+6)));
        else if (instr(var_classe_temp, ' and above') > 0) then
            -- é o último registro
            var_valor_max := 999999999;
        else
            --é um registro intermédio
            var_valor_max := to_number(substr(var_classe_temp,
instr(var_classe_temp, '-')+2));
        end if;
    end if;
        if (not(var_salario_anual > var_valor_max)) then
            exit;
        end if;

    end loop;
    close cursor_classe_salario;
    RETURN var_classe;
end;

/

--Nesta função, é calculado a diferença entre o preço original e o
preço a qual ele foi vendido e calcula-se o desconto
CREATE OR REPLACE FUNCTION "BDII_1010985"."GET_DISCOUNT"
(id_promotion number) return number is
v_prod_id number(6,0);
v_quantity_sold number(3,0);
v_amount_sold number(10,2);
v_prod_list_price number(8,2);
begin
    select quantity_sold, amount_sold, prod_id into v_quantity_sold,
v_amount_sold, v_prod_id from sales where promotion_id=id_promotion
and rownum=1;
    select prod_list_price into v_prod_list_price from products where
prod_id=v_prod_id;
    return 1 - ((v_amount_sold / v_quantity_sold) / v_prod_list_price);
end;

/

--retorna 0 se a data que foi inserida for de manha ou 1 se for de
tarde
CREATE OR REPLACE FUNCTION "BDII_1010985"."GET_MANHA_TARDE"
(DataHora IN date)
return integer
is

begin
    if (to_number(to_char(DataHora, 'hh24')) < 12) then
        return 0;
    else
        return 1;
    end if;
end;

/

```



```

--retorna a data final de uma promoção, é passado por parâmetro o id
da promoção a que se refere
CREATE OR REPLACE FUNCTION "BDII_1010985"."GET_MAX_DATE"
(id_promotion number) return date is
v_date date;
begin
select max(sale_date) into v_date from sales where
promotion_id=id_promotion;
return v_date;
end;

/

--retorna a data inicial de uma promoção, é passado por parâmetro o id
da promoção a que se refere
CREATE OR REPLACE FUNCTION "BDII_1010985"."GET_MIN_DATE"
(id_promotion number) return date is
v_date date;
begin
select min(sale_date) into v_date from sales where
promotion_id=id_promotion;
return v_date;
end;

/

--substitui pontos por vírgulas de um número(em varchar2) passado por
parâmetro
CREATE OR REPLACE FUNCTION "BDII_1010985"."REPLACE_DECIMAL"
(valor_em_texto varchar2)
return number is
begin
return to_number(replace(valor_em_texto, '.', ','));
end;

/

```

Procedimentos

```

--Insere valores na tabela intermédia das promoções e produtos
CREATE OR REPLACE PROCEDURE
"BDII_1010985"."CREATE_DUMMY_PROMO_PRODUCTS" is
v_promotion_id number;
v_prod_id number(6,0);
cursor c_sales is
select prod_id, promotion_id
from sales
where promotion_id is not null
group by (prod_id, promotion_id);--os campos são agrupados para
não haver chaves duplicadas
begin
open c_sales;
loop
fetch c_sales into v_prod_id, v_promotion_id;
exit when c_sales%notfound;
insert into promotions_products values (v_promotion_id,
v_prod_id);

```

```

        end loop;
        close c_sales;
    end;

/

--São inseridos registos na tabela promotions, onde o nome é aleatório
com o prefixo "Promo_"
-- o tipo da promoções também é escolhido aleatoriamente
CREATE OR REPLACE PROCEDURE "BDII_1010985"."CREATE_DUMMY_PROMOS" is
    v_data_inicio date;
    v_data_fim date;
    v_discount number;
    v_loop number;
begin
    v_loop := 1;
    while(v_loop<=34)
    loop
        v_data_inicio := get_min_date(v_loop);
        v_data_fim := get_max_date(v_loop);
        v_discount := get_discount(v_loop) * 100;
        insert into promotions (id_promotion_type, promotion,
date_start, date_end, discount)
            values (dbms_random.value(1, 4),
'Promo_'||dbms_random.string('a', 10), v_data_inicio, v_data_fim,
v_discount);
        v_loop := v_loop+1;
    end loop;
end;

/

--procedimento para inserir na tabela promotion_types
CREATE OR REPLACE PROCEDURE
"BDII_1010985"."CREATE_DUMMY_PROMO_TYPES" is
begin
    insert into promotion_types (promotion_type) values ('Cupões');
    insert into promotion_types (promotion_type) values ('Desconto
direto');
    insert into promotion_types (promotion_type) values ('Cartão
oferta');
    insert into promotion_types (promotion_type) values ('Cartão
cliente');
end;

/

--Procedimento para inserir e actualizar a dimensão canais
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_CHANNELS" is
begin
    insert into BDII_1010834.dim_canais (id_canal, descricao, classe,
custo_venda)
        select upper(channel_id), channel_desc, channel_class,
cost_sale_pct from channels
        minus
        select id_canal, descricao, classe, custo_venda from
BDII_1010834.dim_canais;
    commit;
end;

```

end;

/

--procedimento para inserir na dimensão clientes e na mini dimensão
clientes
--o procedimento procura se os dados a inserir já se encontram na mini
dimensão
--e depois insere na dimensão clientes

```
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_CLIENTES"
IS
CURSOR cursor_mini --cursor para a mini dimensão
IS
SELECT cust.cust_id,
       cust.cust_first_name,
       cust.cust_last_name,
       cust.cust_gender,
       cust.cust_year_of_birth,
       cust.cust_marital_status,
       cust.cust_street_address,
       cust.cust_main_phone_number,
       cust.cust_income_level,
       cust.cust_credit_limit,
       cust.cust_total,
       cust.cust_month_of_birth,
       city.city,
       state.state_province,
       country.country_name,
       sub.country_subregion,
       reg.country_region
FROM   oltp2015.customers@oltp_link_2015 cust
INNER JOIN oltp2015.cities@oltp_link_2015 city
ON      cust.cust_city = city.city_id
INNER JOIN oltp2015.state_provinces@oltp_link_2015 state
ON      city.state_province = state.state_province_id
INNER JOIN oltp2015.countries@oltp_link_2015 country
ON      state.country = country.country_id
INNER JOIN oltp2015.country_subregions@oltp_link_2015 sub
ON      country.country_subregion = sub.country_subregion_id
INNER JOIN oltp2015.country_regions@oltp_link_2015 reg
ON      sub.country_region = reg.country_region_id;
varid oltp2015.customers.cust_id@oltp_link_2015%TYPE;
varfirst oltp2015.customers.cust_first_name@oltp_link_2015%TYPE;
varlast oltp2015.customers.cust_last_name@oltp_link_2015%TYPE;
vargender oltp2015.customers.cust_gender@oltp_link_2015%TYPE;
varyear oltp2015.customers.cust_year_of_birth@oltp_link_2015%TYPE;
varmarital
oltp2015.customers.cust_marital_status@oltp_link_2015%TYPE;
varstreet
oltp2015.customers.cust_street_address@oltp_link_2015%TYPE;
varphone
oltp2015.customers.cust_main_phone_number@oltp_link_2015%TYPE;
varincome oltp2015.customers.cust_income_level@oltp_link_2015%TYPE;
varcredit oltp2015.customers.cust_credit_limit@oltp_link_2015%TYPE;
vartotal oltp2015.customers.cust_total@oltp_link_2015%TYPE;
varmonth oltp2015.customers.cust_month_of_birth@oltp_link_2015%TYPE;
varcity oltp2015.cities.city@oltp_link_2015%TYPE;
varstate
oltp2015.state_provinces.state_province@oltp_link_2015%TYPE;
varcountry oltp2015.countries.country_name@oltp_link_2015%TYPE;
```

```

    varsub
oltp2015.country_subregions.country_subregion@oltp_link_2015%TYPE;
varreg oltp2015.country_regions.country_region@oltp_link_2015%TYPE;
varidade VARCHAR2 (50) ;
minisurr NUMBER;
datan VARCHAR2 (50) ;
nome VARCHAR2 (255) ;
BEGIN
    OPEN cursor_mini;
    LOOP
        FETCH cursor_mini
        INTO varid,
            varfirst,
            varlast,
            vargender,
            varyear,
            varmarital,
            varstreet,
            varphone,
            varincome,
            varcredit,
            vartotal,
            varmonth,
            varcity,
            varstate,
            varcountry,
            varsub,
            varreg;
        EXIT
    WHEN cursor_mini%NOTFOUND;
    varidade :=cal_idade_classe(varyear,varmonth); --vai buscar a
classe correspondente
    IF varmarital IS NULL THEN -- caso de varmarital ser nulo
        varmarital := 'N';
    END IF;
    SELECT COUNT(surr_mini_cliente) --conta o número de registos
iguais na mini dimensão
    INTO minisurr
    FROM BDII_1010834.mini_dim_clientes
    WHERE idade_classe=varidade
    AND genero = vargender
    AND estado_civil =varmarital
    AND salario_classe=varincome
    AND cidade =varcity
    AND provincia =varstate
    AND pais =varcountry
    AND sub_regiao =varsub
    AND regiao = varreg;
    IF minisurr <1 THEN -- Caso de não haver registos na mini
dimensão
        INSERT
        INTO BDII_1010834.mini_dim_clientes
        (
            idade_classe,
            genero,
            estado_civil,
            salario_classe,
            cidade,
            provincia,
            pais,
            sub_regiao,

```

```

        regiao
    )
VALUES
(
    varidade,
    vargender,
    varmarital,
    varincome,
    varcity,
    varstate,
    varcountry,
    varsub,
    varreg
);
-- Vai buscar o último registo inserido na mini dimensão
SELECT MAX(surr_mini_cliente) INTO minisurr FROM
BDII_1010834.mini_dim_clientes;
ELSE
    SELECT surr_mini_cliente --Caso exista, vai buscar a surrogate
correspondente
    INTO minisurr
    FROM BDII_1010834.mini_dim_clientes
    WHERE idade_classe=varidade
    AND genero      = vargender
    AND estado_civil = varmarital
    AND salario_classe=varincome
    AND cidade      = varcity
    AND provincia   = varstate
    AND pais        = varcountry
    AND sub_regiao   = varsub
    AND regiao      = varreg;
END IF;
datan      := varyear || '/' || varmonth;--Converte o ano e o mes de
nascimento para um único campo
nome       := varfirst || ' ' || varlast;--Converte o primeiro e
último nome para um único campo
vartotal   := replace_decimal(vartotal);--Substitui o ponto por
vírgula e retorna o campo em formato número
INSERT
INTO BDII_1010834.dim_clientes
(
    id_cliente,
    data_nascimento,
    limite_credito,
    custo_total,
    fk_mini_cliente,
    nome
)
VALUES
(
    varid,
    datan,
    varcredit,
    vartotal,
    minisurr,
    nome
);

END LOOP;
COMMIT;
CLOSE cursor_mini;

```

```

END;

/

-- procedimento para inserir dados nos facto compras
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_COMPRAS"
IS
BEGIN
    INSERT
    INTO BDII_1010834.fac_compras
    (
        surr_tempo,
        surr_empregado,
        surr_produto,
        surr_fornecedor,
        custo_unitario,
        preco_unitario,
        quantidade,
        custo_total,
        preco_total
    )
    SELECT dim_tempo.surr_tempo,
        dim_empregados.surr_empregado,
        dim_produtos.surr_produto,
        dim_fornecedores.surr_fornecedor,
        buys_oltp.unit_cost,
        buys_oltp.unit_price,
        SUM(buys_oltp.quantity_bought),
        SUM(buys_oltp.unit_cost * buys_oltp.quantity_bought),
        SUM(buys_oltp.unit_price * buys_oltp.quantity_bought)
        --junta as dimensões e as tabelas da bd operacional num único
inner join
    FROM oltp2015.buys@oltp_link_2015 buys_oltp
    INNER JOIN bdii_1010834.dim_produtos dim_produtos
    ON dim_produtos.id_produto = buys_oltp.prod_id
    INNER JOIN BDII_1010834.dim_fornecedores dim_fornecedores
    ON dim_fornecedores.id_fornecedor = buys_oltp.supplier_id
    INNER JOIN bdii_1010834.dim_tempo dim_tempo
    ON dim_tempo.dia = to_number(TO_CHAR(buys_oltp.buy_date,
'DD'))
    AND dim_tempo.mes = to_number(TO_CHAR(buys_oltp.buy_date,
'MM'))
    AND dim_tempo.ano = to_number(TO_CHAR(buys_oltp.buy_date,
'YYYY'))
    AND dim_tempo.de_manha = get_manha_tarde(buys_oltp.buy_date) --ve se
a data corresponde a manha ou tarde
    INNER JOIN BDII_1010834.dim_empregados dim_empregados
    ON dim_empregados.id_empregado = buys_oltp.employee_id
    GROUP BY ( --os dados são agrupados para não existirem chaves
duplicadas
        dim_tempo.surr_tempo,
        dim_empregados.surr_empregado,
        dim_produtos.surr_produto,
        dim_fornecedores.surr_fornecedor,
        buys_oltp.unit_cost,
        buys_oltp.unit_price
    );
    COMMIT;
END;

```

/

```
--procedimento para carregar a dimensão empregados
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_EMPREGADOS" is

begin
  insert into BDII_1010834.dim_empregados
  (
    id_empregado,
    comissao,
    nome,
    data_contrato,
    trabalho,
    chefe,
    salario_classe
  )
  select emp.employee_id,
    emp.commission_pct,
    emp.first_name || ' ' || emp.last_name,
    emp.hire_date,
    emp.job_id,
    chefe.first_name || ' ' || chefe.last_name,
    get_classe_salario(emp.salary)
  --criou-se um inner join com a própria tabela de modo a obter os
  nomes dos chefe dos empregados
  from BDII_1010832.employees chefe
  right join BDII_1010832.employees emp
  on chefe.employee_id = emp.manager_id
  ;
  commit;
end;
```

/

```
--procedimento para carregar os factos promoções
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_FAC_PROMOCOES"
IS
  error_code      number;
  error_msg       varchar2(255) ;
BEGIN
  INSERT
  INTO BDII_1010834.fac_promocoes
  (
    surr_produto,
    surr_tempo,
    surr_promocao
  )
  SELECT dim_produtos.surr_produto,
    dim_tempo.surr_tempo,
    dim_promocoes.surr_promocao
  FROM oltp2015.products@oltp_link_2015 products
  INNER JOIN promotions_products
  ON promotions_products.id_product = products.prod_id
  INNER JOIN promotions
  ON promotions.id_promotion = promotions_products.id_promotion
  JOIN bdii_1010834.dim_tempo dim_tempo
  ON to_date(dim_tempo.dia
    || '/'
    || dim_tempo.mes
```

```

        || '/'
        || dim_tempo.ano, 'dd/mm/yyyy') BETWEEN promotions.date_start AND
promotions.date_end
    INNER JOIN bdii_1010834.dim_produtos dim_produtos
    ON dim_produtos.id_produto = products.prod_id
    INNER JOIN bdii_1010834.dim_promocoes dim_promocoes
    ON dim_promocoes.id_promocao = promotions.id_promotion
    MINUS
    SELECT * FROM bdii_1010834.fac_promocoes ;
    COMMIT;
EXCEPTION--se ocorrer erro, guarda o registo numa tabela à parte
WHEN OTHERS THEN
    error_code := SQLCODE;
    error_msg := SUBSTR(SQLERRM, 1, 200);
    INSERT
    INTO log_errors
    (
        ERROR_CODE ,
        ERROR_MSG ,
        OPERATION_NAME,
        OPERATION_DESC
    )
    VALUES
    (
        error_code,
        error_msg,
        'load_vendas',
        'Execução do procedimento load_vendas'
    );
END;

/

```

```

--procedimento para inserir e actualizar fornecedores
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_FORNECEDORES" is
begin
    insert into BDII_1010834.dim_fornecedores (id_fornecedor, nome,
endereco)
    select supplier_id, name, address
    from oltp2015.suppliers@oltp_link_2015
    minus
    select id_fornecedor, nome, endereco
    from BDII_1010834.dim_fornecedores;
    commit;
end;

/

```

```

--procedimento para inserir produtos
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_PRODUTOS" is
begin
    insert into BDII_1010834.dim_produtos (id_produto, categoria,
sub_categoria, classe_peso, unidade_medida
        , tamanho_pacote, estado, lista_preco, min_preco, custo_prod,
descricao)
    select prod_oltp.prod_id, cat_oltp.prod_category,
sub_cat_oltp.prod_subcategory, prod_oltp.prod_weight_class
        , prod_oltp.prod_unit_of_measure, prod_oltp.prod_pack_size

```



```

        , prod_oltp.prod_status, prod_oltp.prod_list_price,
prod_oltp.prod_min_price
        , prod_oltp.prod_cost, desc_oltp.prod_desc
    from oltp2015.products@oltp_link_2015 prod_oltp
    inner join oltp2015.sub_categories@oltp_link_2015 sub_cat_oltp
    on sub_cat_oltp.sub_cat_id = prod_oltp.sub_cat_id
    inner join oltp2015.categories@oltp_link_2015 cat_oltp
    on cat_oltp.cat_id = sub_cat_oltp.cat_id
    inner join oltp2015.product_descriptions@oltp_link_2015
desc_oltp
    on desc_oltp.prod_desc_id = prod_oltp.prod_descriptions_id;
    commit;
end;

/

```

```

--procedimento para carregar promoções
CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_PROMOCOES"
IS
    var_count NUMBER; -- n° de registos da tabela dim_promocoies
BEGIN
    SELECT COUNT(*) INTO var_count FROM BDII_1010834.dim_promocoies;
    IF (var_count = 0) THEN--carregamento do registo 0 caso a dimensão
    promoções não tenha registos
        INSERT
        INTO BDII_1010834.dim_promocoies
        (
            id_promocao,
            nome,
            desconto,
            tipo_promocao,
            data_inicio,
            data_fim
        )
        VALUES
        (
            0,
            'Sem promoção',
            0,
            'Sem tipo',
            to_date('1.1.1900', 'dd.mm.yyyy'),
            to_date('1.1.1900', 'dd.mm.yyyy')
        );
    END IF;
    INSERT
    INTO BDII_1010834.dim_promocoies
    (
        id_promocao,
        nome,
        desconto,
        tipo_promocao,
        data_inicio,
        data_fim
    )
    SELECT promotions.id_promotion,
    promotions.promotion,
    promotions.discount,
    promotion_types.promotion_type,
    promotions.date_start,
    promotions.date_end

```

```

FROM promotions
INNER JOIN promotion_types
ON promotions.id_promotion_type = promotion_types.id_promotion_type
MINUS
SELECT dw_promocoos.id_promocao,
       dw_promocoos.nome,
       dw_promocoos.desconto,
       dw_promocoos.tipo_promocao,
       dw_promocoos.data_inicio,
       dw_promocoos.data_fim
FROM BDII_1010834.dim_promocoos dw_promocoos;
COMMIT;
END;

```

/

--procedimento para carregar a dimensao tempo entre a data minima e a data maxima da tabela de vendas

```

CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_TEMPO"
is
  v_ano_minimo number(4);
  v_ano_maximo number(4);
  v_data_temp date;
begin
  --seleciona a data minima e maxima da tabela de vendas
  select to_number(to_char(min(SALES.SALE_DATE), 'yyyy')) into
v_ano_minimo from SALES;
  select to_number(to_char(max(SALES.SALE_DATE), 'yyyy')) into
v_ano_maximo from SALES;
  v_data_temp := to_date(to_char('01-01-'||v_ano_minimo), 'dd-mm-
yyyy');
  WHILE v_data_temp < to_date(to_char('31-12-'||v_ano_maximo), 'dd-mm-
yyyy')
  LOOP--insere registros entre as datas
    INSERT INTO SYN_TEMPO (Dia, Mes, Ano, Dia_Semana, De_Manha)
    VALUES (
      to_number(to_char(v_data_temp, 'dd')),
      to_number(to_char(v_data_temp, 'mm')),
      to_number(to_char(v_data_temp, 'yyyy')),
      to_char(v_data_temp, 'Day', 'NLS_DATE_LANGUAGE=Portuguese'),
      0
    );
    INSERT INTO SYN_TEMPO (Dia, Mes, Ano, Dia_Semana, De_Manha)
    VALUES (
      to_number(to_char(v_data_temp, 'dd')),
      to_number(to_char(v_data_temp, 'mm')),
      to_number(to_char(v_data_temp, 'yyyy')),
      to_char(v_data_temp, 'Day', 'NLS_DATE_LANGUAGE=Portuguese'),
      1
    );
    v_data_temp := v_data_temp+1;
  END LOOP;
  commit;
end;

```

/

--procedimento para carregar as vendas

```

CREATE OR REPLACE PROCEDURE "BDII_1010985"."LOAD_VENDAS" is

```

```

var_surr_empregado number;
var_surr_promocao number;
error_code         number;
error_msg          varchar2(255);

begin

select surr_empregado
into var_surr_empregado
from bdii_1010834.dim_empregados
where id_empregado=0;
select surr_promocao
into var_surr_promocao
from bdii_1010834.dim_promocoes
where id_promocao=0;
insert into BDII_1010834.fac_vendas (
    surr_tempo,
    surr_canal,
    surr_cliente,
    surr_promocao,
    surr_produto,
    surr_mini_cliente,
    surr_empregado,
    unid_vendidas,
    custo_compra,
    valor_venda,
    lucro)
    select dim_tempo.surr_tempo,
    dim_canais.surr_canal,
    dim_clientes.surr_cliente,
    nvl(dim_promocoes.surr_promocao, var_surr_promocao),--caso haja
nulls,é inserido o registro extra
    dim_produtos.surr_produto,
    dim_clientes.fk_mini_cliente,
    nvl(dim_empregados.surr_empregado, var_surr_empregado),--caso haja
nulls,é inserido o registro extra
    sum(sales_oltp.quantity_sold),
    sum(dim_produtos.custo_prod * sales_oltp.quantity_sold),
    sum(sales_oltp.amount_sold),
    sum(sales_oltp.amount_sold - (dim_produtos.custo_prod *
sales_oltp.quantity_sold))
from oltp2015.sales@oltp_link_2015 sales_oltp
inner join bdii_1010834.dim_clientes dim_clientes
on dim_clientes.id_cliente = sales_oltp.cust_id
inner join bdii_1010834.dim_produtos dim_produtos
on dim_produtos.id_produto = sales_oltp.prod_id
inner join bdii_1010834.dim_canais dim_canais
on dim_canais.id_canal = sales_oltp.channel_id
inner join bdii_1010834.dim_tempo dim_tempo
on dim_tempo.dia = to_number(to_char(sales_oltp.sale_date, 'DD'))
and dim_tempo.mes = to_number(to_char(sales_oltp.sale_date, 'MM'))
and dim_tempo.ano = to_number(to_char(sales_oltp.sale_date,
'YYYY'))
and dim_tempo.de_manha = get_manha_tarde(sales_oltp.sale_date)
left join bdii_1010834.dim_empregados dim_empregados
on dim_empregados.id_empregado = sales_oltp.employee_id
left join bdii_1010834.dim_promocoes dim_promocoes
on dim_promocoes.id_promocao = sales_oltp.promotion_id
group by
    dim_tempo.surr_tempo,
    dim_canais.surr_canal,

```

```

dim_clientes.surr_cliente,
dim_promocoes.surr_promocao,
dim_produtos.surr_produto,
dim_clientes.fk_mini_cliente,
dim_empregados.surr_empregado
minus
select * from bdii_1010834.fac_vendas
;
commit;
EXCEPTION-- em caso de erro, é registado numa tabela à parte
when others then
    error_code := SQLCODE;
    error_msg  := SUBSTR(SQLERRM, 1, 200);

    insert into log_errors (
        ERROR_CODE      ,
        ERROR_MSG       ,
        OPERATION_NAME,
        OPERATION_DESC
    ) values (
        error_code,
        error_msg,
        'load_vendas',
        'Execução do procedimento load_vendas'
    );
end;
/

```

Sinónimos

```

CREATE OR REPLACE SYNONYM "BDII_1010985"."CHANNELS" FOR
"BDII_1010832"."CHANNELS";

CREATE OR REPLACE SYNONYM "BDII_1010985"."PRODUCTS" FOR
"OLTP2015"."PRODUCTS"@"OLTP_LINK_2015.REGRESS.RDBMS.DEV.US.ORACLE.COM";

CREATE OR REPLACE SYNONYM "BDII_1010985"."SALES" FOR
"OLTP2015"."SALES"@"OLTP_LINK_2015.REGRESS.RDBMS.DEV.US.ORACLE.COM";

CREATE OR REPLACE SYNONYM "BDII_1010985"."SYN_TEMPO" FOR
"BDII_1010834"."DIM_TEMPO";

```

Computador C (Vasco)

Criação da ligação à base de dados operacional

```
create database link oltp_link_query
connect to oltp_query identified by oltp_query
using 'oltp';
```

Sequências

```
create sequence pk_canais;

create sequence pk_clientes;

create sequence pk_employees;

create sequence pk_mini_clientes;

create sequence pk_produtos;

create sequence pk_promocoes;

create sequence pk_suppliers;

create sequence pk_tempo;

create sequence pk_vendas;
```

Tabelas

```
create table Dim_Tempo(
  Surr_Tempo number NOT NULL,
  Dia number(2,0) NOT NULL,
  Mes number(2,0) NOT NULL,
  Ano number(4,0) NOT NULL,
  Dia_Semana varchar2(255) NOT NULL,
  De_Manha number(1,0) NOT NULL
);

create table Dim_Promocoes(
  Surr_Promocao number NOT NULL,
  Id_Promocao number NOT NULL,
  Nome varchar2(255) NOT NULL,
  Desconto number NOT NULL,
  Tipo_Promocao varchar2(255) NOT NULL,
  Data_Inicio date NOT NULL,
  Data_Fim date NOT NULL
);

create table Dim_Canais (
  Surr_Canal number NOT NULL,
  Id_Canal number NOT NULL,
  Descricao varchar2(255) NOT NULL,
  Classe varchar2(255) NOT NULL,
  Custo_Venda number(4,3) NOT NULL
);
```

```

create table Dim_Produtos (
Surr_Produto number NOT NULL,
Id_Produto number NOT NULL,
Categoria varchar2(255) NOT NULL,
Sub_Categoria varchar2(255) NOT NULL,
Classe_Peso number NULL,
Unidade_Medida varchar2(255) NULL,
Tamanho_Pacote varchar2(255) NULL,
Estado varchar2(255) not null,
Lista_Precos number NOT NULL,
Min_Precos number NOT NULL,
Custo_prod number null,
Descricao varchar2(255) NULL
);

```

```

create table Dim_Clientes(
Surr_Cliente number NOT NULL,
Id_Cliente number NOT NULL,
Data_Nascimento date NOT NULL,
Limite_Credito number(20) NOT NULL,
Custo_Total number NOT NULL,
Fk_Mini_cliente number NOT NULL,
Nome varchar(255) NOT NULL
);

```

```

create table Mini_Dim_Clientes(
Surr_mini_cliente number NOT NULL,
Idade_classe varchar2(255) NOT NULL,
Genero varchar2(255) NOT NULL,
Estado_Civil varchar2(255) NOT NULL,
Salario_Classe number NOT NULL,
Cidade varchar2(255) NOT NULL,
Provincia varchar2(255) NOT NULL,
Pais varchar2(255) NOT NULL,
Sub_Regiao varchar2(255) NOT NULL,
Regiao varchar2(255) NOT NULL
);

```

```

create table Fac_Vendas (
Surr_Tempo number NOT NULL,
Surr_Canal number NOT NULL,
Surr_Cliente number NOT NULL,
Surr_Promocao number NOT NULL,
Surr_Produto number NOT NULL,
Surr_mini_cliente number NOT NULL,
Surr_empregado number NOT NULL,
Unid_Vendidas number NOT NULL,
Custo_Compra number NOT NULL,
Valor_Venda number NOT NULL,
Lucro number not null
);

```

```

create table Dim_Empregados (
Surr_Empregado number NOT NULL,
Id_Empregado number NOT NULL,
Comissao number NULL,
Nome varchar2(255) NULL,
Data_contrato date null,
Trabalho varchar2(255) NULL,
Chefe varchar2(255) NULL,

```

```
Salario_Classe varchar2(255) NULL
);
```

```
create table Dim_Fornecedores(
Surr_Fornecedor number NOT NULL,
Id_Fornecedor number NOT NULL,
Nome varchar2(255) NOT NULL,
Endereco varchar2(255) NOT NULL
);
```

```
create table Fac_Compras(
Surr_Tempo number NOT NULL,
Surr_Empregado number NOT NULL,
Surr_Produto number NOT NULL,
Surr_Fornecedor number NOT NULL,
Custo_Unitario number NOT NULL,
Preco_Unitario number NOT NULL,
Quantidade number NOT NULL,
Custo_Total number NOT NULL,
Preco_Total number NOT NULL
);
```

```
create table Fac_Promocoas(
Surr_Produto number NOT NULL,
Surr_Tempo number NOT NULL,
Surr_Promocao number NOT NULL
);
```

Index's

Constraints

```
ALTER TABLE Dim_Tempo
ADD PRIMARY KEY (Surr_Tempo);
```

```
ALTER TABLE Dim_Promocoas
ADD PRIMARY KEY (Surr_Promocao);
```

```
ALTER TABLE Dim_Canais
ADD PRIMARY KEY (Surr_Canal);
```

```
ALTER TABLE Dim_Produtos
ADD PRIMARY KEY (Surr_Produto);
```

```
ALTER TABLE Dim_Clientes
ADD PRIMARY KEY (Surr_Cliente);
```

```
alter table Dim_Empregados
add primary key (Surr_Empregado);
```

```
ALTER TABLE Dim_Fornecedores
ADD PRIMARY KEY (Surr_Fornecedor);
```

```
ALTER TABLE Mini_Dim_Clientes
ADD PRIMARY KEY (Surr_mini_cliente);
```

```
ALTER TABLE Fac_Vendas
```

```
ADD PRIMARY KEY
(Surr_Tempo,Surr_Canal,Surr_Promocao,Surr_Produto,Surr_Cliente,Surr_mi
ni_cliente,Surr_empregado);
```

```
ALTER TABLE Fac_Compras
ADD PRIMARY KEY
(Surr_Tempo,Surr_Empregado,Surr_Produto,Surr_Fornecedor);
```

```
ALTER TABLE Fac_Promocoes
ADD PRIMARY KEY (Surr_Produto,Surr_Tempo,Surr_Promocao);
```

Constraints de referência

```
ALTER TABLE Dim_clientes
ADD CONSTRAINT FK_Clientes_mini_clientes
FOREIGN KEY (Fk_mini_cliente)
REFERENCES mini_dim_clientes(surr_mini_cliente);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_Tempo
FOREIGN KEY (Surr_Tempo)
REFERENCES Dim_Tempo(Surr_Tempo);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_Promocoes
FOREIGN KEY (Surr_Promocao)
REFERENCES Dim_Promocoes(Surr_Promocao);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_Canais
FOREIGN KEY (Surr_Canal)
REFERENCES Dim_Canais(Surr_Canal);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_Clientes
FOREIGN KEY (Surr_Cliente)
REFERENCES Dim_Clientes(Surr_Cliente);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_Produtos
FOREIGN KEY (Surr_Produto)
REFERENCES Dim_Produtos(Surr_Produto);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_Empregados
FOREIGN KEY (Surr_Empregado)
REFERENCES Dim_empregados(Surr_empregado);
```

```
ALTER TABLE Fac_Vendas
ADD CONSTRAINT FK_Vendas_mini_clientes
FOREIGN KEY (Surr_mini_cliente)
REFERENCES mini_dim_clientes(surr_mini_cliente);
```

```
ALTER TABLE Fac_Compras
ADD CONSTRAINT FK_Compras_Tempo
FOREIGN KEY (Surr_Tempo)
REFERENCES Dim_Tempo(Surr_Tempo);
```

```
ALTER TABLE Fac_Compras
```



```

ADD CONSTRAINT FK_Compras_empregados
FOREIGN KEY (Surr_Empregado)
REFERENCES Dim_Empregados(Surr_Empregado);

ALTER TABLE Fac_Compras
ADD CONSTRAINT FK_Compras_Produtos
FOREIGN KEY (Surr_Produto)
REFERENCES Dim_Produtos(Surr_Produto);

ALTER TABLE Fac_Compras
ADD CONSTRAINT FK_Compras_Fornecedores
FOREIGN KEY (Surr_fornecedor)
REFERENCES Dim_fornecedores(Surr_fornecedor);

ALTER TABLE Fac_Promocoes
ADD CONSTRAINT FK_Promocoes_Tempo
FOREIGN KEY (Surr_Tempo)
REFERENCES Dim_Tempo(Surr_Tempo);

ALTER TABLE Fac_Promocoes
ADD CONSTRAINT FK_Promocoes_Produto
FOREIGN KEY (Surr_Produto)
REFERENCES Dim_Produtos(Surr_Produto);

ALTER TABLE Fac_Promocoes
ADD CONSTRAINT FK_Promocoes_Promocao
FOREIGN KEY (Surr_Promocao)
REFERENCES Dim_Promocoes(Surr_Promocao);

```

Permissões

```

grant select,insert,update,delete on dim_canais to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on dim_clientes to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on mini_dim_clientes to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on dim_empregados to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on dim_produtos to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on dim_promocoes to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on dim_fornecedores to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on dim_tempo to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on fac_compras to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on fac_promocoes to
BDII_1010985,BDII_1010832;
grant select,insert,update,delete on fac_vendas to
BDII_1010985,BDII_1010832;

```

Triggers

```

create or replace TRIGGER trigger_surr_canais
before insert on dim_canais
for each row
begin

```

```

    select pk_canais.nextval into :new.surr_canal from dual;
end;
/

create or replace TRIGGER trigger_surr_tempo
before insert on dim_tempo
for each row
begin
    select pk_tempo.nextval into :new.surr_tempo from dual;
end;
/

create or replace TRIGGER pk_empregados
before insert on dim_empregados
for each row

begin
    select pk_employees.nextval into :new.surr_empregado from dual;
end;
/

create or replace TRIGGER trigger_surr_promocoes
before insert on dim_promocoes
for each row
begin
    select pk_promocoes.nextval into :new.surr_promocao from dual;
end;
/

create or replace TRIGGER trigger_surr_fornecedores
before insert on dim_fornecedores
for each row
begin
    select pk_suppliers.nextval into :new.surr_fornecedor from dual;
end;
/

create or replace TRIGGER trigger_surr_produtos
before insert on dim_produtos
for each row
begin
    select pk_produtos.nextval into :new.surr_produto from dual;
end;
/

create or replace TRIGGER trigger_surr_mini_cliente
before insert on mini_dim_clientes
for each row
begin
    select pk_mini_cliente.nextval into :new.surr_mini_cliente from
dual;
end;
/

```

Registos complementares

```

insert into dim_empregados values
(0,0,0,'Desconhecido',null,'Desconhecido',null,'Desconhecido');
insert into dim_promocoes values (0,0,'Sem Promoção',0,'Sem Tipo');

```