

Projeto Final

SCC0541 - Laboratório de Bases de Dados

Kelvin Guilherme de Oliveira - 9293286

Lucas Yudi Sugi - 9293251

Mauricio Caetano da Silva - 9040996

INTRODUÇÃO

O projeto consiste em implementar um sistema para a empresa *Accessories and Bike Company* (ABC) onde será possível realizar simulações, gerar relatórios e ter uma noção geral de como anda o funcionamento da empresa, permitindo aos gerentes melhores tomadas de decisões.

OBJETIVO

O objetivo deste documento é abordar o desenvolvimento do trabalho, focando nos seguintes pontos:

- Atendimento dos requisitos propostos;
- Dificuldades encontradas;
- Soluções implementadas;
- Testes;
- Demonstração da aplicação;

Para saber executar e testar o sistema de maneira adequada, consulte a seção “*REQUISITO Extra - Teste*”.

REQUISITO 1 - Controle de acesso ao sistema

Para controlar o acesso ao sistema, foi utilizada a abordagem de controle na aplicação, ou seja, todo o controle está no Java, e não na base de dados. Este requisito é alcançado inserindo funções que verificam qual o tipo do usuário que está logando no sistema (tipo 1, tipo 2, tipo 3 ou tipo 4), e, de acordo com o tipo do usuário, libera e bloqueia as funcionalidades desejadas.

Os usuários do tipo 1 apenas podem gerar relatórios, enquanto os do tipo 2 podem gerar relatórios e fazer simulações, os do tipo 3 podem usufruir de todas as funcionalidades do sistema, e, por fim, os do tipo 4 não têm acesso ao sistema.

Além do controle de usuário, em cada login, o sistema inclui uma nova tupla na tabela “SYS_LOG”, criada na base de dados, contendo um ID único, o ID do usuário que está logado, e o horário que o mesmo acessou o sistema. O acesso a esses dados não é disponibilizado pela aplicação, sendo necessária a consulta direta no banco de dados.

A seção que demonstra como testar o sistema detalha 4 usuários cadastrados no sistema (um de cada tipo) para verificar tal funcionalidade.

REQUISITO 2 - Relatórios

Nesta seção será explicado como as consultas foram implementadas para atender os requisitos indicados. É importante salientar que para maiores detalhes é necessário consultar o arquivo *relatório.sql*. Nas seguintes subseções abordaremos apenas em um ponto de vista abstrato.

1. Clientes cujo cartão de crédito já venceu

Para obter um relatório dos clientes cujo cartão de crédito já venceu utilizamos a cláusula *where* que é responsável por comparar a data de vencimento do cartão com a data atual.

Desse modo, quando temos que o ano de vencimento é menor que o ano atual, ou caso ambos sejam iguais mas o mês atual ultrapasse o mês de vencimento, realizamos uma extração da tupla.

Com as linhas extraídas é realizado um agrupamento do cliente para que todas suas compras fiquem em um único grupo. Dessa forma, é possível retornar a data da sua última venda utilizando a função de agregação *max*. Por fim, são projetados os dados de interesse para o usuário.

2. Histórico de departamento dos funcionários

Obtemos um relatório do histórico de departamento de cada funcionário realizando as junções necessárias nas tuplas para possuir o conteúdo adequado. Tendo as informações necessárias é realizado uma projeção com os dados mais importantes para o usuário.

O problema de nulos na tabela é resolvido na projeção utilizando a expressão `case` que permite “converter” o valor `null` para `Ativo`.

3. Dados de frete

O relatório sobre dados de frete é realizado por meio de várias subconsultas. Na primeira é efetuado um agrupamento pelos anos das vendas o que permite obter a sua correspondência do total de frete. Além disso, na projeção utilizamos a expressão `case` para separar as vendas em dois grupos (`total > 2000`, `total <= 2000`) que são representados por 1 e 0, respectivamente.

Assim, obtemos uma tabela em que uma linha possui o ano, total de frete e identificador de grupo. Para converter esses dados em linha, já que, a tabela pode possuir algo como “linha 1: 2018 | 1000 | 1”, “linha 2: 2018|2000|2” utilizamos a função `pivot` que permite gerar a seguinte configuração: “linha 1: 2018|1000|2000”.

Como nessa nova tabela é possível obter valores nulos, pois um ano pode ter uma venda de apenas um grupo, uma nova consulta é realizada projetando as informações com a expressão `case` que “converte” `null` à 0.

Por fim, com a intenção de obter o total de frete no ano é realizado uma projeção em que há uma soma das colunas dos dois grupos.

4. Vendas de cada ano

Provemos um relatório de vendas utilizando várias subconsultas. No primeiro `select` realizamos um agrupamento por ano e meses obtendo seu respectivo total de vendas.

Assim, o resultado é uma tabela com a seguinte configuração: “linha 1: 2018|1|500”, “linha 2: 2018|2|600”. Note que com isso obtemos os resultados necessários, porém em um aspecto visual que não é tão interessante para o usuário.

Desse modo, utilizamos a função pivot que permite transpor os meses que estão em colunas para linhas, além disso, ela é responsável por fazer a correspondência 1 -> Janeiro, 2 -> Fevereiro, e assim por diante.

Logo, obtemos uma tabela em que temos na primeira coluna o ano e nas subsequentes os meses que estão preenchidas com o seu total de vendas. Como é possível existir valores nulos nessa configuração, uma nova consulta é realizada em conjunto com a expressão case, para projetar valores nulos à 0.

Por fim, temos mais dois select's sendo o primeiro responsável por somar o valor de todos os meses (por meio da projeção) e o segundo por calcular o percentual de alteração das vendas por ano por meio da função LAG que permite efetuar operações em linha na tabela.

5. 15 produtos mais vendidos no semestre

O relatório dos 15 produtos mais vendidos no semestre também utiliza várias subconsultas. Inicialmente é realizado um agrupamento por ano e mês para obter um produto que foi vendido nessas datas (no arquivo relatorio.sql há um agrupamento por mais campos do produto, mas isso não afeta o resultado final pois um produto igual possui a mesma configuração, já que, ele é baseado em seu identificador).

Além disso, na projeção dessa consulta é utilizada a expressão case que é responsável por verificar se o mês está no primeiro ou segundo semestre, gerando uma coluna com os valores 1 e 0 para os respectivos grupos.

Sabendo quais são os semestres e anos de cada produto nas vendas, uma nova consulta é realizada para gerar uma coluna que indicará a posições dos produtos mais vendidos, por exemplo, a linha "Bearing Ball,1,2018,2" indica que o produto Bearing Ball no semestre 1 do ano de 2018 é o segundo (2) produto mais comprado. Isso é efetuado por meio da função row_number.

Possuindo indicadores de posição baseado em um grupo de ano e semestre, uma última consulta é realizada em conjunto com a cláusula where que é responsável por retornar os 15 produtos mais vendidos de cada conjunto.

6. Pares de produtos vendidos juntos numa mesma venda

O relatório de pares de produtos mais vendidos utiliza várias subconsultas da mesma forma que nos requisitos anteriores. Primeiramente é realizado um select para obter uma tabela em que possuímos como principal dado o identificador da venda e seu respectivo produto nela.

Como uma venda pode possuir vários produtos, uma nova consulta é realizada para projetar uma coluna que indicará a posição do item segundo o seu valor total, ou seja, se o produto Bearing Ball é o terceiro com maior valor numa venda então ele terá o valor 3 nesta coluna.

Desse modo, sabendo as posições de cada produto nas suas vendas, um select é efetuado na finalidade de obter apenas os dois itens com maior valor (utiliza a cláusula where).

Assim, temos uma tabela que apresenta os pares de produtos mais vendidos na venda. O problema nessa configuração é que os dados estão dispostos em linhas, i.e, o par de produtos está organizado em duas linhas.

Logo, para transpor esse resultado em colunas, utilizamos a função listagg que será responsável por unir o resultado separado por um traço ('-'), ou seja, temos algo como: "linha 1: 1|Bearing Ball-Blade|Bikes-Bikes" que significa, na venda de identificador 1 os pares de produto são Bearing Ball, Blade e suas categorias respectivas são Bikes e Bikes(as outras colunas foram omitidas).

Note que o resultado necessário para o usuário está presente nessa tabela, contudo o mesmo não se encontra de uma maneira visual adequada, pois é necessário realizar uma correspondência no valor das células de uma coluna com outra coluna.

Desse modo, uma nova consulta é realizada para projetar os valores da célula em colunas, ou seja, realizaremos um split. Isso é possível utilizando as funções substr e instr que auxiliam a separar a string presente na célula.

Note nesse ponto que é possível obter vendas em que há apenas um produto vendido, ou seja, não existe um par. Para resolver isso um select é realizado com a cláusula where na qual verifica onde existe produtos nulos, "excluindo" essas linhas da tabela.

O resultado dessa etapa é uma tabela com todas as informações necessárias para o usuário. Porém, novamente, a configuração não é adequada pois os pares de produtos mais vendidos estão em vendas separadas, por exemplo, “linha 1: 1|Bearing Ball|Blade”, “linha 2: 2|Blade|Bearing Ball”.

Dessa maneira, existe a necessidade de unir esses valores em um único grupo, mas existe o problema dos pares estarem em linhas trocadas conforme o exemplo anterior, o que não seria reconhecido como mesmo grupo pela sintaxe group by.

Portanto, uma consulta é feita junto com a expressão case que irá redistribuir os valores nessa coluna de modo que o menor identificador do produto fique na primeira coluna e o maior na segunda. Por fim, realizamos o agrupamento necessário junto com as funções de agregação para obter a quantidade e total dos pares de produtos nas vendas em que somente ambos aparecem juntos.

7. Vendas por país

Para obter o relatório das vendas dos países inicialmente é feita uma consulta que agrupa segundo ano, país e estado mostrando seu o total vendido. Além disso, na projeção utilizamos a função row_number para atribuir identificadores de posição ordenados pela venda segundo um ano e país, ou seja, criamos uma coluna que indica qual a posição de um estado considerando um ano e país que é organizado pela total das vendas.

Assim, temos informações de quais são os estados que possuem maiores valores de total. Tendo esses dados, uma nova consulta é efetuada para selecionar o maior estado (utiliza a cláusula where). Além disso, um agrupamento é feito para permitir somar o total de todos os estados (função sum) e obter o maior valor total do estado (função max).

Note que para retornar o nome do maior estado dentro do grupo é utilizada a função first_value.

Após essas etapas temos uma tabela que indica o ano, país, estado, total do estado e total do país. Perceba, portanto, que é possível haver vários valores de anos para um mesmo país e estado, como o usuário deseja ver apenas o último

ano, uma nova consulta é realizada para projetar (utilizando row_number) uma coluna que conterá valores indicando qual o último ano.

Por fim, um select final é feito para selecionar apenas as linhas que possuem na última coluna de identificação o valor 1 que representa o último ano do país e estado.

8. Observações

Note que neste ponto todas as consultas da parte de relatório foram explicadas, porém em nenhuma delas ocorreu uma citação de como é realizada a filtragem.

Isso porque todas elas utilizam a mesma estratégia: filtrar com a cláusula where. Assim, quando se deseja saber algo específico da consulta o where é usado para buscar tal(is) linha(s).

A única diferença entre elas é que a cláusula é colocada no meio das subconsultas em posições estratégicas, ou seja, onde é possível remover mais linhas para que haja um ganho de desempenho.

REQUISITO 3 - Simulações

Para realizar as simulações propostas foi utilizado as materialized views que permitem criar cópias das tabelas base para cada usuário. Tais visões foram criadas utilizando o modo fast, permitido pela oracle.

Assim, quando um usuário deseja realizar uma simulação é criado uma view na qual é associado ao nome dela o identificador único do usuário, por exemplo, supondo que José possua id 10, então é criado a view com o seguinte nome VIEW_PRODUTO_10 (visão da tabela produto).

Isso é necessário pois cada usuário deve possuir sua própria tabela para realizar as simulações.

Para que seja possível desfazer as alterações feitas na tabela é utilizado o método DROP que exclui a view criado para o usuário.

REQUISITO 4 - Telas

As telas do sistema foram desenvolvidas em Java, utilizando Java.Swing. Todas as telas foram projetadas para permitir um melhor aproveitamento das funcionalidades disponibilizadas pelo sistema, tendo uma tela de Login, uma Home, uma para geração de relatórios, uma para ter um overview, outra para fazer simulações, e, por fim, uma tela que permite a visualização dos relatórios.

Para fazer a tela de Overview, diversas consultas precisaram ser feitas na base de dados. Tais consultas estão explicadas abaixo.

1. Total vendido no dia, mês e ano

Para essa consulta foi utilizada unicamente a tabela VENDA, na qual foram extraídos os valores totais das vendas realizadas em um certo período. Os funcionários dos tipos 2 e 3 podem realizar essas consultas determinando uma data específica, com dia, mês e ano, ou então utilizar um mês e um ano ou ainda consultar os valores de venda apenas por um ano.

2. Top-3 Funcionários que mais venderam no mês e ano

Nessa consulta foram utilizadas as tabelas VENDEDOR, PESSOA e VENDA, para que pudessem ser extraídos os valores totais de vendas realizadas pelos vendedores da empresa em um determinado período (mês ou ano), além dos nomes desses funcionários.

3. Top-15 produtos mais vendidos, juntamente com a quantidade

Os 15 produtos mais vendidos foram encontrados analisando todo o histórico contido no banco de dados por meio das tabelas ITEMVENDA, VENDA e PRODUTO. Esses produtos foram agrupados por nome conforme apareciam nas vendas e então

4. Clientes que mais compraram no ano

A lógica seguida para essa consulta é muito próxima da lógica utilizada na consulta dos 15 produtos mais vendidos, descrita acima, com a diferença que nesse

caso analisamos as tabelas CLIENTE, PESSOA E VENDA, observando o número de vezes que um cliente estava associado a uma venda em um ano específico.

5. Clientes que mais compraram

A consulta realizada para esse caso também segue a mesma ideia da consulta anterior, para clientes que mais compraram em um ano, com a diferença que nesta foi analisado todo o histórico de compras contido no banco de dados e não apenas em um ano específico.

6. Nome e quantidade dos produtos cujo estoque é menor que 10

Essa consulta utiliza a tabela PRODUTO, observando os valores do atributo quantidade. Desse modo foram filtrados os produtos com estoque menor que 10 unidades.

REQUISITO Extra - Teste

Para que o sistema da empresa ABC funcione corretamente são realizadas diversas consultas, tanto para geração de relatórios quanto para as telas de visão geral da empresa. Dentro desse contexto, utilizamos a base de dados antiga adaptada para que todas as consultas retornem um resultado consistente.

A adaptação foi feita por meio da diminuição da quantidade de tuplas da base para a maioria das tabelas (PESSOA, VENDAS, PRODUTOS, etc) visando uma maior velocidade no desenvolvimento do sistema. Além disso alguns valores foram modificados para facilitar a manutenção dessa base: foram trocadas as chaves primárias e estrangeiras de valor grande para valores pequenos, foram ajustados os valores de frete, valores totais de venda, valores de cada produto, peso de produtos, etc.

Além disso, para facilitar o acesso ao sistema, foram criados 4 usuários adicionais, um de cada tipo, que são:

- Tipo1: ***kelvin.oliveira@usp.br/123456.***
- Tipo2: ***lucas.sugi@usp.br/12345678.***
- Tipo3: ***neymarjesus@usp.br/hexa123.***

- Tipo4: ***mauricio@usp.br/12345678***.

Sendo assim, para executar o programa, deve-se acessar o diretório do programa (*PROJETO_FINAL*), depois acessar o subdiretório */gui* e rodar o seguinte comando via terminal: ***java -jar ABC.jar***.

Com o programa em execução, deve-se logar com um dos 4 usuários listados acima, e pode-se verificar o controle de acesso de cada usuário. Supondo um usuário de acesso ilimitado (tipo 3), pode-se ter um Overview da empresa, gerar relatórios específicos e até mesmo fazer as simulações. Vale ressaltar que, em cada vez que a simulação é feita, os relatórios e o overview tomará como base a simulação feita, e não mais a tabela original.

Um único detalhe da interface gráfica é que, quando deseja-se deslogar de um usuário e entrar com uma nova conta, deve-se entrar na área de Simulações antes de fazer qualquer outra operação, apenas para garantir o sucesso na execução do programa. Para encerrar o programa, o usuário pode sair de sua conta (deslogar) e fechar o sistema, no botão fechar (X).