

MÓDULO 1: Exercícios SQL

PARTE 1:

- 1) Fazer uma pesquisa que retorne o código de atendimento e a data de atendimento e a data de alta.

Tabelas: atendime. **(Exercício resolvido na aula 6).**

- 2) Fazer um Select que traga o código de atendimento, nome do paciente, nome do convênio e somente dos atendimentos do 02/05/2016. Obs: Dependendo do banco, pode não haver retornos. Tabelas: atendime, paciente e convenio. **(Exercício resolvido na aula 6).**

- 3) Criar uma pesquisa que informe o nome do paciente, o código do paciente, o nome da cidade da cidade em que o paciente reside, e o código de atendimento do paciente.

Tabelas: atendime, cidade e paciente.

- 4) Fazer um Select dos pacientes internados durante o mês de agosto de 2016, com idade maior ou igual a 50 anos. Dica: o resultado terá 20 retornos (linhas). **(Exercício resolvido na aula 6).**

- 5) Total de atendimento do mês de janeiro de 2018, separados por tipo de atendimento. **(Exercício resolvido na aula 7).**

- 6) Prestador com mais cirurgias em 2017. Tabelas: atendime, aviso_cirurgia e prestador. **(Exercício resolvido na aula 7).**

- 7) Retornar as 10 cirurgias mais realizadas em 2017.

- 8) Um select traga a idade do paciente utilizando o comando sysdate e data de nascimento, anexando na pesquisa a CID (Classificação Internacional de Doenças), para informar a descrição da doença que foi indicada no atendimento do paciente.

- 9) Criar uma pesquisa para atendimentos de código maior que 5000 cujo o código do prestador do atendimento seja 407, 1562, 3071. Dica: Utilizar o relacional IN.

- 10) Fazer um select com todos os atendimentos realizados por prestadores do estado do Rio de Janeiro.

PARTE 2:

- 1) Fazer um select com o somatório de prestadores por estado de residencia. Dica: utilizar SUM() e GROUP BY.

- 2) Fazer um select com código de todos os itens da conta de internação, nome do paciente e nome do procedimento. Tabelas: itreg_fat, pro_fat, paciente. **(Exercício resolvido na aula 7).**

- 3) Fazer um select com todos os códigos das contas com mais de R\$1000,00 no mês de setembro de 2017. Trazer códigos da conta e valor, ordenados por valor decrescente.

- 4) Quantidade de pacientes atendidos no ano de 2016.

- 5) Fazer um select com o somatório de faturamento em internação por prestador no ano de 2018. Dica: agrupar pelo prestador o somatório de valores unitários.

- 6) Fazer um select com a quantidade do exame CULTURA PARA FUNGOS no ano de 2017, agrupadas por prestador. Dica: utilizar a itreg_fat, pro_fat (tabela dos procedimentos) e prestador.

PARTE 3:

- 1) Trazer numa linha só o paciente com mais atendimento em 2018, com o código, nome do paciente e convênio. Dica: fazer um contador para o número de atendimentos.
- 2) Fazer um select com todas conta da itreg_fat, nome do paciente, nome do prestador, nome do convenio, nome do procedimento, valor do procedimento (vl_conta_total) dos 3 primeiros meses de 2017. **(Exercício resolvido na aula 11).**
- 3) Fazer um select igual ao anterior, mas agora para as contas ambulatoriais, que utilizam a reg_amb e itreg_amb.

PARTE 4:

- 1) Trazer os itens das contas ambulatoriais e de internação fechadas, com nome do convenio, nome do paciente e valor unitário e valor total. Dica: Utilizar a coluna sn_fechada da reg_amb e reg_fat.
- 2) Idem, mas agora trazer um somatório por convênio.
- 3) Média do tempo de internação (paciente que tiveram internação, tempo entre entrada e alta) por faixa etária (0 a 10, 11 a 18, 19 a 30, 30 a 45, 46 a 60, 60 a 80 e maior que 81). **(Exercício resolvido na aula 11).**

MÓDULO 2: Exercícios DML

- 1) Inicialmente, vamos criar uma tabela com o nome do aluno, com todos pacientes cadastrados no ano de 2017, no formato:

```
CREATE TABLE PC_TREINA_MATHEUS AS
SELECT * FROM PACIENTE
WHERE TRUNC(HR_CADASTRO,'YYYY') = TO_DATE('01/01/2017', 'DD/MM/YYYY');
```

Posteriormente, vamos:

- Deletar todos pacientes que começam com a letra C;
- Depois, mudar o estado civil para 'C' de todos pacientes da cidade SÃO PAULO CAPITAL;
- Por fim, inserir todos pacientes cadastrados no ano de 2019.

(Exercício resolvido na aula 16).

- 2) Neste exercício, vamos criar uma tabela de backup da tabela produto, com todos produtos cadastrados em 2015, com nome prod_treina_nome:

- Deletar todos produtos da subclasse 16;
- Depois, atualizar a hora da última entrada para o SYSDATE em todos os produtos considerados consignados.

- 3) Criar uma cópia da tabela pro_fat (formato pro_fat_treina_nome) com todos os procedimentos que estão presentes na tabela cirurgia:

- Deletar todos os procedimentos que precisam de 3 ou mais auxiliares;
- Depois, alterar o cd_gru_pro (grupo de faturamento) 52 para 99;
- Por fim, importar da tabela pro_fat todos procedimentos com cd_gru_pro = 1.

(Exercício resolvido na aula 16).

- 4) Criar uma cópia da tabela prestador (formato presta_treina_nome), com todos os prestadores ativos:
- Deletar todos prestadores que o conselho é CRM;
 - Depois, inativar todos prestadores da cidade do RIO DE JANEIRO;
 - Por fim, copiar todos os prestadores inativos da tabela prestador.

MÓDULO 2: Exercícios PL/SQL

PARTE 1: FUNCTION

- 1) Criar uma function que retorne última data de atendimento de um paciente.

Premissas:

- Ter como parâmetro de entrada o cd_paciente;
- Ter como saída (retorno) o maior dt_atendimento.

- 2) Criar uma function traga o valor total atual de uma conta ambulatorial, recalculando todo os valores unitários.

Premissas:

- Buscar os itens da conta na ITREG_AMB;
- Fazer o somatório de todos os intes na coluna vl_unitario;
- Ter como parametro de entrada o numero da conta (cd_reg_amb).

(Exercício resolvido na aula 21).

- 3) Criar uma function semelhante ao exercício anterior, mas para as contas de internação (ITREG_FAT);

- 4) Criar uma ÚNICA função que retorne o valor atual de uma conta, tendo como parâmetro de entrada o numero da conta e o tipo de conta.

Premissas:

- Esse segundo parametro vai informar o tipo de conta, se é ambulatorial ou internação, e desta forma, se deverá utilizar a itreg_fat ou itreg_amb.

- 5) Criar uma função que retorne o numero do laudo de um exame.

Premissas:

- Ter como parâmetro de entrada o CD_ITPED_RX.
- CD_LAUDO pode ser encontrado na ITPED_RX;
- Caso não exista o laudo (cd_laudo é null) deve retornar uma mensagem com LAUDO NÃO EMITIDO.

- 6) Criar uma function que retorne o valor da emissão de uma carteira para o paciente.

Premissas:

- Ter como parâmetros de entrada o cd_paciente e cd_convenio, e ter como retorno o valor da carteira a ser cobrado;
- Conferir se a carteira para o paciente e convenio existe na tabela CARTEIRA;
- Caso não exista, é 1º via, logo deve retornar o valor de R\$20,00.
- Caso já exista a carteira, então será a 2º via, e o valor deve ser R\$35,00.

(Exercício resolvido na aula 21).

PARTE 2: PROCEDURE

- 1) Inicialmente, vamos criar uma tabela de backup da tabela reg_fat e itreg_fat, com nome do aluno no formato:

```
CREATE TABLE ITREG_FAT_MATHEUS AS  
SELECT * FROM ITREG_FAT;  
CREATE TABLE REG_FAT_MATHEUS AS  
SELECT * FROM REG_FAT
```

Depois disto, vamos criar uma procedure que atualize NA TABELA CÓPIA o vl_total_conta, somando todos os vl_unitario e fazendo o update na COPIA DA ITREG_FAT e REG_FAT criada com o nome do aluno.

Premissas:

- Ter como parâmetro de entrada o numero da conta.

- 2) Para essa procedure, vamos criar uma copia da tabela PACIENTE, com nome do aluno no formato:

```
CREATE TABLE PACIENTE_MATHEUS AS  
SELECT * FROM PACIENTE;
```

Depois disto, vamos deletar todos os paciente desta tabela de cópia, deixando-a vazia.

Agora vamos criar uma procedure que recebe apenas uma letra, e importa da tabela paciente todos os pacientes com aquela letra para tabela cópia.

Premissas:

- Utilizar um cursor e um loop, fazendo o insert de uma linha por vez;
- Ter como parâmetro de entrada a primeira letra do nome.

- 3) Vamos criar uma tabela de backup da tabela EST_PRO, com nome do aluno no formato:

```
CREATE TABLE EST_PRO_MATHEUS AS  
SELECT * FROM EST_PRO
```

Depois disto, vamos criar uma procedure que altera as quantidades de um produto na tabela de cópia. Logo, terá como parâmetro de entrada o cd_produto, o cd_estoque e a nova quantidade.

Premissas:

- Caso não exista esse produto, deverá fazer o insert na EST_PRO do produto para aquele cd_estoque;

(Exercício resolvido na aula 24).

- 4) Nesta atividade, vamos criar uma procedure que insere um procedimento na conta, consultando o valor do procedimento na val_pro, inserindo o procedimento e recalculando os valores totais. UTILIZAR A COPIA DA TABELA ITREG_FAT E REG_FAT DO EXERCÍCIO 1.

A procedure deverá atender as seguintes premissas:

- Criar procedure com nome PRC2_AL_WL_NOMEDOALUNO;
- Deve ter como parametro de entrada o CD_PRO_FAT e o numero da conta.

VAL_PRO	ITREG_FAT
vl_honorario	vl_honorario_unitario
vl_operacional	vl_operacional_unitario
vl_total	vl_unitario

- Buscar na tabela VAL_PRO o as seguinte colunas que deverão ser inseridas na ITREG_FAT_NOME_DO_ALUNO.
- Recalcular o valor total da conta e atualizar o vl_total_conta com novo valor;

- Buscar na tabela gru_fat o cd_gru_fat correspondente aquela pro_fat;
- Outros valores serão os seguintes: cd_setor_produziu = 855, cd_setor = 855, cd_lancamento = (ultimo valor + 1) e cd_usuario = nome_do_aluno;

PARTE 3: TRIGGERS

- 1) Neste exercício, vamos utilizar a cópia da tabela PACIENTE, já criada no banco de dados com o nome de cada aluno, que foi utilizada no exercício procedure. Iremos criar uma trigger de log que registre as alterações que ela recebe.

Premissas:

- Trigger deve ter o nome TRG_PAC_NOMEDOALUNO;
- O primeiro passo é criar uma tabela de log, igual a tabela utilizada;
- Depois, adicionar as colunas TP_DML (para o tipo de alteração, se é delete, insert ou update), data_alteracao e usuario;
- Por fim, criar a trigger propriamente dita;

- 2) Criar uma trigger que não permite a alteração em uma conta fechada. Para isso, utilizar as cópias das ITREG_FAT e REG_FAT das tabelas criadas no exercício 1 de de Procedure.

Premissas:

- Trigger deve estar na cópia da tabela ITREG_FAT;
- Deve fazer a checagem se a conta está fechada, na coluna sn_fechada na cópia da REG_FAT;
- Caso a conta esteja fechada, deve disparar um RAISE_APPLICATION_ERROR, com a mensagem NÃO É PERMITIDO INSERIR, EXCLUIR OU ALTERAR ITENS DE UMA CONTA FECHADA;

- 3) Nestes exercício, vamos utilizar a cópia da tabela EST_PRO, já criada no banco de dados com o nome de cada aluno, que foi utilizada no exercício procedure. Iremos criar uma trigger de log que registre as alterações que ela recebe.

Premissas:

- Registrar todas as operações de UPDATE que a quantidade antiga e nova tem uma diferença maior que 5;
- Não permitir UPDATE que a diferença entre a quantidade antiga e nova seja maior que 10.

(Exercício resolvido na aula 24).

Desenvolvidos por Matheus Lutz Ramos

Sulwork, 2022