

PráticaSQL(PG)-Comandos

Apresentação, instalação e outras dicas.

Este documento é um exercício sobre o SQL:1992. Você PODE testá-lo em qualquer SGBD relacional. Aqui fizemos nossos testes em um SGBD PostgreSQL, versão 9.2.

Este documento apresenta um exercício **VÁLIDO PARA NOTA**, este trabalho deve ser feito em partes ou etapas. Este trabalho é INDIVIDUAL, embora tenha uma etapa que pode ser feito de forma colaborativa para acelerar o ritmo do desenvolvimento.

Por favor, preencha o nome e matrícula do aluno no espaço abaixo disponível.

Nome e matrícula do aluno:

Matrícula	Nome
-----------	------

DEPOIS de resolvido o SQL(PG)-CMD TROQUE o nome do arquivo para: **SQL(PG)-NomeDoAluno.pdf** e envie-o para a conta de e-mail divulgada no site da disciplina.

No assunto escreva: **IBDX-SQL(PG)-NomeDoAluno**, com X=**N** ADS Noturno ou X=**T** ADS Tarde. Escreva SEU **NomeDoAluno** como no exemplo: *José da Silva* → **JoseDaSilva**
Ou de outra forma:

- USE a primeira letra de cada parte do nome com maiúsculas e o resto com minúsculas e TODO o nome SEM espaços em branco e SEM acentos.
- NÃO use apelido ou partes abreviadas do nome, **ESCREVA** como está na sua matrícula.

Ambiente de Trabalho.

O ambiente de software para executar este trabalho deverá ter:

- Servidor de Página: Usa-se o Apache (versão 2.0 ou mais recente), ainda é possível usar o TomCat ou o IIS.
- Interpretador de comandos da linguagem PHP
- Servidor de Banco de Dados: Usa-se o SGBD PostgreSQL (versão 9.2 ou mais recente).
- pgAdmin 4 (versão 3.6 ou mais recente).

Neste projeto recomenda-se o pacote XAMPP versão mais recente. Este pacote instala tudo o que é necessário e é mais fácil de desinstalar (em ambiente Windows) sem deixar o compactador com muitos subprogramas espalhados no equipamento.

Neste projeto não se usa o MySQL (ou o MariaDB). Ao instalar o XAMP escolha fazer a instalação somente do servidor de página, a linguagem PHP e o banco dados (MariaDB ou MySQL), mas neste projeto não usaremos o Banco de Dados do pacote XAMP. Usaremos o Servidor de Banco de Dados **PostgreSQL**.

Depois de baixado o pacote XAMPP instale-o seguindo as instruções de instalação padrão. O procedimento coloca o XAMPP no diretório **C:\XAMPP**.

Eu aconselho instalar o XAMPP e depois o PostgreSQL. Seu computador fica com dois SGBDs. O PostgreSQL se instala como serviço do Windows mas o MySQL só é configurado para isso se você quiser. Use o painel de controle do XAMP e deixe configurado para o APACHE ser iniciado juntamente com o Windows.

Para o PHP conseguir acessar o PostgreSQL edite o arquivo PHP.INI que fica no diretório C:\XAMPP\PHP (use um editor de texto simples tipo notetab ou notepad). Neste arquivo procure a linha: **;extension=phppgsql.dll**, tire o ponto-e-vírgula do início da linha, ela fica escrita assim: extension=phppgsql.dll.

Grave a alteração. Reinicie seu computador (em algumas máquinas o reinício não é necessário, em outras é obrigatório, por isso: reinicie seu computador).

Pronto para este trabalho é o que precisamos: do Apache, PHP e o PostgreSQL rodando.
Bem, agora começa a carga de trabalho.

O que devemos fazer?

Este trabalho é dividido em 6 (SEIS) partes:

1. Acerto do Dicionário de Dados da base (acesse o PGSQL com o aplicativo PgAdmin 4 - vem no instalador do PostgreSQL, ou pode ser atualizado separadamente).
2. Tradução de Álgebra Relacional → SQL (são apresentadas perguntas e a AR já especificada, seu trabalho: Escrever os comandos SQL).
3. Escrita da AR e SQL (Você recebe as questões e deve desenvolver a AR e os comandos SQL correspondentes – dica: não precisa ser uma SQL muito sofisticada).
4. Escrita de SQL (você recebe a questão e deve apresentar o SQL que resolve o problema – dica: faça a AR, depois traduza para SQL e depois otimize).

Estas partes devem ser feitas na ordem que estão apresentadas, **acredite**, é mais produtivo e efetivo desenvolver este trabalho cumprindo cada parte inteiramente antes de partir para seguinte.

Você deverá desenvolver uma base com 101 tabelas e duas visões!

Para acelerar o trabalho será disponibilizado um script em SQL que cria umas... 80 tabelas (ou um pouco mais)...

O Acerto do dicionário de dados da base **DEVE SER FEITO considerando CERTO** o dicionário que está disponibilizado no site da disciplina.

De outro modo, a Base de Dados que vocês receberam está com **DEFEITOS**, ora falta campo em tabela ora a definição de um campo está errada ou então falta uma tabela inteira ou tem tabela que não precisa estar na base de dados.

Novamente afirmo: o que está CORRETO é o que está escrito neste documento. Portanto vocês devem acertar a base de dados.

Este é um trabalho extenso. Faça-o com cuidado e atenção.

Vamos ao trabalho!

Acerto do Dicionário de Dados (no SGBD PostgreSQL)

Modelo de dados

Todos os modelos de dados foram desenvolvidos com a notação do Modelo Relacional. Todos os modelos estão disponibilizados no site da disciplina. Muitas tabelas da base já apresentam dados (estão *populadas*), durante os ajustes das tabelas tente manter os dados nas tabelas.

Dicionário de dados

Para agilizar a implementação da base de dados, no site da disciplina existe um acesso para o download de um script que gera muitas das tabelas desta base de dados (**GERADOR.SQL**). Estas tabelas (geradas pelo script) NÃO estão como especificadas neste dicionário de dados. Por conta disso o trabalho sobre esta base deve ser realizado em etapas:

- Crie uma base de dados com nome **LBDPG**. Faça o download do Script **GERADOR.SQL** e execute o script (criando as tabelas iniciais do exercício)
- Faça os ajustes de estrutura em todas as tabelas desta base. **Note**: os campos devem estar definidos como neste dicionário de dados. Entretanto, não necessariamente na mesma ordem como os campos aparecem nestas tabelas. Ou seja, os campos não precisam estar na mesma ordem como está no dicionário. Nesta etapa do trabalho aproveite e defina as CHAVES PRIMÁRIAS das tabelas.
- Depois de ajustadas as tabelas (estrutura e Chaves Primárias) faça a definição das Chaves Estrangeiras. NOTE: cada par de campos PK e FK deve ser analisado em separado para determinar o comportamento das chaves estrangeiras quando se atualizam os valores das correspondentes chaves primárias (em caso de DELETE ou UPDATE).

Neste dicionário de dados tentou-se seguir a seguinte notação: **Negrito** é Chave Primária, *Itálico* é Chave Estrangeira, Chave Candidata não tem notação, somente são explicadas no campo **comentário**. Se ao executar a análise do Dicionário você perceber qualquer imperfeição na definição de campos das tabelas avise o professor. A alteração será analisada e atualizada no Dicionário (sendo destacada em cor **AZUL**) e uma nova data passará a constar na estrutura de menus que dá acesso ao TesteSQL.

Todo o Dicionário de Dados está disponibilizado no site da disciplina. Deste modo eventuais atualizações serão percebidas mais rapidamente.

Este dicionário de dados apresenta **DIFERENÇAS** em relação ao que foi desenvolvido na base LBDPG. Considere tudo o que está neste dicionário como **CERTO**. Portanto, acerte as estruturas de TODAS as tabelas de sua base de dados PostgreSQL.

NOTE: é fundamental que todos os campos estejam definidos nas tabelas correspondentes, entretanto a ORDEM dos campos na tabela não é relevante.

Nesta parte do trabalho (ajuste do dicionário de dados) aconselho que seja feito em **dupla**.

Enquanto um trabalha editando os campos das tabelas o outro fica com o dicionário aberto e verificando o andamento do trabalho de acerto da base. Isso pode ser feito até que se conclua o acerto do dicionário de dados da Base.

DEPOIS que a base estiver ajustada pode-se inserir linhas nas tabelas com o script **INSERT.SQL**, depois criam-se as CPs e CEs.

Aqui TERMINA a parte do trabalho que pode ser feita em dupla.

Faça uma cópia da base (formato PLAIN com DATA, PREDATA e POSDATA, INSERT COMMAND) e passe para o ser par.

A partir deste ponto o trabalho DEVE ser individual. Terminadas todas estas tarefas iniciam-se a Prática dos Comandos.

Prática de Comandos SQL

Usando o pgAdmin 4, acesse a base de dados do exercício. Acesse a lista de tabelas e abra uma janela “QUERY TOOL” para execução dos comandos. A cada comando você deve usar esta janela para testar os comandos que pesquisar para tentar resolver a proposta da pergunta.

Como vocês estão trabalhando com a versão PDF aconselho que os comentários sejam escritos usando o programa FOXIT (<http://cdn01.foxitsoftware.com/pub/foxit/reader/desktop/win/4.x/4.3/enu/FoxitReader431enuSetup.exe>). Este programa permite colocar janela de texto sobre os PDF. Se você conhece outro que faça isso, tudo bem. Use-o, mas saiba que seu trabalho será corrigido com o programa FOXIT, ou seja:

- **TESTE seu SQL de resposta antes de me enviar.**

A Prática de comandos tem três partes:

- Tradução de AR para SQL – Você recebe uma pergunta, a AR que resolve a pergunta e deve montar a lista de comandos que é a tradução da AR para SQL. Nesta parte podem acontecer que o SQL imponha alterações na AR pelo fato do SGBD ser ou não limitado no tratamento da tradução do SQL. POR ISSO, recomenda-se fazer a alteração necessária e depois escrever a sequência de SQL que implementa a AR.
- Desenvolvimento de AR e SQL. – Aqui o aluno recebe somente a pergunta e deve escrever a AR e depois traduzi-la para SQL.
- Perguntas e Comandos de Resposta. – Aqui a proposta é analisar uma pergunta e escrever UM COMANDO em SQL que consiga resolver a pergunta.

Vocês devem testar estes comandos tanto em PostgreSQL. ACONSELHO que vocês façam o trabalho abrindo uma aba com o pgAdmin4, o arquivo PDF onde vai escrever as resposta e UM editor de programas. Analise a pergunta, escreva a resposta no editor de programas (teste sua resposta no pgAdmin 4), depois copie/cole a resposta no arquivo PDF. MANTENHA o PDF sempre salvo a cada resposta escrita.

Bem, vamos aos comandos em SQL.

Tradução Álgebra Relacional → SQL

Nota-se que: Pode haver alguma diferença no SQL dos dois

Agora que sabemos o que os comandos do SQL podem processar, vamos traduzir a Álgebra Relacional para comandos da SQL. As perguntas estão feitas e a Álgebra Relacional que apresenta a resposta está formulada. Você deve traduzir as expressões de AR para SQL.

Sugestão: traduza cada expressão da AR para um comando SQL. Teste sua resposta no banco de dados (comando por comando, e a seguir todos de uma vez). Sabemos que o SQL pode combinar em um comando mais de uma expressão da AR. **Se você quiser otimizar** o código SQL pode fazê-lo, **MAS** deixe na resposta a sua tradução expressão por expressão.

Para este contexto fizemos a especificação da Álgebra Relacional para resolver a algumas perguntas. Vamos apresentar as perguntas, as respostas em AR e a seguir você pratica a tradução para SQL.

```
1.    Quais ônibus fizeram viagens em todas as rotas viárias no ano de 2010?
-- Preparando...SEL : A ← viagens [[(dtsaidaviagem entre "2010-01-01" e "2010-12-31") OU
                                (dtchegadaviagem entre "2010-01-01" e "2010-12-31")]]
                                PRO : B ← A [[pkviagem,fkrota]]
                                JUN : C ← B [[ B.pkviagem=passagens.fkviagem ]] passagens
                                PRO : PD ← C [[fkrota, fkonibus]](distintos)
                                PRO : P ← rotasviarias [[ pkrota→fkrota ]]
                                PRO : F ← onibus [[pkonibus→fkonibus ]]
-- Dividindo    PCT : S ← P x F
                                SUB : T ← S - PD
                                PRO : W ← T [[ fkonibus ]]
                                SUB : R ← F - W
-- Completando JUN : Z <- R [[ R.fkonibus=onibus.pkonibus ]] onibus
                                PRO : Z [[ukplaca,txapelido, nuanofabricacao, qtcapacidade]]
```

Em SQL:

2. Quais os nomes dos professores e os nomes das respectivas disciplinas atribuídas aos professores que ministram disciplinas da Área de Estudo "EXATAS"?

```
SEL : A ← areadeestudo[[txnomearea="EXATAS"]]
JUN : B ← A[[A.pkareaestudo = K.fkareaestudos]] areaestudodisciplinas →K
PRO : B1 ← B[[fkdisciplinas]]
JUN : C ← B1[[B1.fkdisciplinas = disciplinas.pkdisciplina]]disciplinas
PRO : C1 ← C[[pkdisciplina→pkdisc, txnome→txnomedisciplina]]
JUN : D ← C1[[C1.pkdisc = atribuicoes.fkdisciplina]]atribuicoes
PRO : D1 ← D[[fkprofessor, txnomedisciplina]]
JUN : E ← D1[[D1.fkprofessor = professores.pkprofessor]]professores
PRO : R ← E[[txnomeprofessor,txnomedisciplina]]
```

Em SQL:

3. Qual é a quantidade de disciplina atribuídas a cada professor?

```
PRO : A ← Atribuicoes[[CONTA(fkprofessor)→qtddisc,fkprofessor]]
                        {Agrupado por fkprofessor}
                        {Ordenado por fkprofessor}
JUN : B ← A[[A.fkprofessor = Pr.pkprofessor]]Professores→Pr
PRO : B[[qtddisc,pkprofessor,txnomeprofessor]]
```

Em SQL:

4. Quais são os nomes dos professores que ministram todas disciplinas (note que uma parte da resposta especifica o procedimento da divisão)?

Preparando o procedimento da divisão

Projeção: P $\leftarrow$ professores[[pkprofessor $\rightarrow$ idprof]]

Projeção: D $\leftarrow$ disciplinas[[pkdisciplina $\rightarrow$ iddisc]]

Projeção: PD $\leftarrow$ atribuições[[fkprofessor $\rightarrow$ idprof, fkdisciplina $\rightarrow$ iddisc]]

Estas três tabelas ficam prontas para o procedimento da divisão

Procedimento da Divisão:

A $\leftarrow$ P X D - Esta tab. Fica UC com PD (atribuições)

B $\leftarrow$ A - PD - A é maior que Atribuições

C $\leftarrow$ B[[idprof]] - Aqui temos os prof. Que não se ligam a pelo menos uma disciplina

E $\leftarrow$ P - C - Aqui tenho os códigos de professores que ministram TODAS as disc.

Conclusão da Pergunta

Junção...: F $\leftarrow$ E[[E.idprof=Professores.pkprofessor]]Professores

Projeção: F[[pkprofessor, txnomeprofessor]]

Em SQL:

5. Qual é a média da quantidade de disciplinas atribuídas aos professores para cada área de estudo? O que pode representar esta média (interprete o resultado)

W $\leftarrow$ atribuiçoes [[fkprofessor $\rightarrow$ idprof, fkdisciplina $\rightarrow$ iddisc]]

S $\leftarrow$ W [[W.iddisc = disciplinas.pkdisciplina][disciplinas

T $\leftarrow$ S [[idprof,pkdisciplina $\rightarrow$ iddisc]]

K $\leftarrow$ T [[T.iddisc= aed.fkdisciplinas]] areaestudodisciplinas $\rightarrow$ aed

M $\leftarrow$ K [[media(idprof) $\rightarrow$ QtdMedProf, fkareaestudos]] {agrupar por fkareaestudos} {ordenar por fkareaestudos}

L $\leftarrow$ M[[K.fkareaestudos=areaestudo.pkareaestudo]]areaestudo

L[[txnomearea,QtdMedProf]]

Em SQL:

6. Quantas disciplinas são atribuídas aos professores que ministram disciplinas da Área de Estudo "Humanas"?

A ← `areaestudo[[txnomearea="Humanas"]]`

B ← `A[[ A.pkareaestudo = K.fkareaestudos ]]areaestudodisciplinas → K`

C ← `B[[ fkdisciplinas ]]`

D ← `C[[ C.fkdisciplinas = disciplinas.pkdisciplinas ]]disciplinas`

E ← `D[[ pkdisciplina → iddisc ]]`

F ← `E[[ E.iddisc = atribuicoes.fkdisciplina ]]atribuicoes`

`F[[fkdisciplina , conta(fkdisciplina) ]] {agrupa por fkprofessor} {ordem por fkprofessor}`

Em SQL:

7. Quais são, em ordem decrescente, as disciplinas com mais professores atribuídas?

A ← `atribuicoes [[ fkdisciplina, conta(fkprofessor) → qtdprof ]]`

`{agrupa por fkdisciplina tendo conta(fkprofessor)>3}`

`{ordem por conta(fkprofessor) decrescente}`

B ← `A [[ A.fkdisciplina = disciplinas.pkdisciplina ]] disciplinas`

B `[[pkdisciplina, txnome, qtdprof, txementa, qthoras, txcritérioavaliacao, dtcaddisciplina]]`

Em SQL:

8. Quais são todas as rotas viárias que passam pelas cidades "São Paulo" e "Rio de Janeiro"?

```
A <- cidades [[ txnomecidade="São Paulo" ou txnome="Rio de Janeiro" ]]  
B1 <- A [[ A.pkcidade = rotasviarias.fkcidadeorigem]] rotasviarias  
B2 <- A [[ A.pkcidade = rotasviarias.fkcidadedestino]] rotasviarias  
C <- B1 U B2  
C [[ pkrota, txnomerota, txdescrperiodo, fkcidadeorigem , fkcidadedestino ]]
```

Em SQL:

9. Quais são os nomes de funcionários que passaram por consultas de todas as especialidades médicas nos últimos 180 dias

```
a <- consultas [[ dthoraconsulta >= hoje-180 ]]  
b <- a [[ fkfuncionario, fkmedico ]]  
c <- b [[ b.fkmedico = medicos.pkmedico ]] medicos  
PD<- c [[ fkfuncionario→idfunc, fkespecialidade→idespec ]]  
P <- funcionarios [[ pkfuncionario→idfunc ]]  
D <- especmedicas [[ pkespecialidade→idespec ]]
```

```
S <- P X D  
T <- S - PD  
W <- T [[ idfunc ]]  
R <- P - W
```

```
V <- R[[R.idfunc=f.pkfuncionario]]funcionários→f  
V[[txprenomes,txsobrenome]]
```

Em SQL:

Resolução de Perguntas com Álgebra Relacional e SQL

Agora podemos implementar os comandos que respondem a algumas perguntas sobre esta base de dados.

Faça esta etapa da seguinte maneira: Analise a pergunta e formule os comandos em Álgebra relacional que respondem a pergunta. Depois de formulada a resposta em AR, escreva os comandos em SQL usando o máximo possível de aproximação com a Álgebra Relacional. Se desejar usar alguma técnica da linguagem SQL para melhorar o comando faça COMENTÁRIOS explicando sua implementação.

1. Quais foram os funcionários (cite: primeiro nome e sobrenome) que participaram nos projetos feitos pelo departamento onde o gerente é o funcionário "HEATHER"?

2. Quais foram os nomes das ações que foram tomadas em projetos que são dos departamentos onde o departamento superior é o departamento "A01"?

3.	Quais são os nomes dos funcionários que registraram vendas para os clientes da cidade de "Bauru"?

4. Quais são os nomes dos projetos que utilizaram todas as tarefas?

5. Quantas notas fiscais de vendas foram emitidas para cada cliente no primeiro semestre de 2016? Para cliente sem nota fiscal emita o valor "ZERO"

6. Quantos funcionários atuaram em todos os projetos no ano de 2002?

7. Exiba os nomes das Instituições de ensino que oferecem mais do que 5 cursos de capacitação na Área de Estudo: "Ciência da Computação"

8. Exiba os títulos de livros do acervo de livros que NÃO foram movimentados no ano de 2018 e calcule a percentagem destes livros em relação ao total de livros do acervo. O que pode representar este número?

9. Qual a percentagem de funcionários que NÃO tem carro?

10. Qual a quantidade de consultas realizadas no ano de 2010 que foi coberta por planos de saúde? Qual a percentagem de funcionários que usaram estas consultas? Se o custo por consulta é de R\$40,00 e os planos de saúde custam R\$350,00 por funcionário, responda: Qual o número mínimo de consultas que torna o pagamento dos planos de saúde com uma relação custo/benefício positivo?

Perguntas e Comandos de Resposta

1. A base de dados apresentada tem a necessidade de ser atualizada com alguns registros para ter efeito alguns comandos que serão escritos mais adiante neste exercício. Se as chaves estrangeiras das suas tabelas estiverem certas, execute este comando para ajustar os valores das Chaves Primárias da tabela PRODUTOS. Note que no modelo de dados esta tabela apresenta relacionamento com pedcomprasitens, pedvendasitens, nfcomprasitens e nfvendasitens. Se suas tabelas não estiverem certas, os dados de produtos nestas tabelas não devem ter sido alterados. Se isso aconteceu você deve efetuar as alterações com o comando UPDATE nas tabelas erradas. NESTE caso, escreva abaixo o comando que você escreveu para ajustar os valores de produtos na tabela que apresentou o erro. SE o erro NÃO ocorreu, sei que você definiu corretamente o valor das Chaves Estrangeiras.

```
update produtos SET pkproduto=(case when pkproduto='001003001002' then 1
                                when pkproduto='001001001001' then 2
                                when pkproduto='001001001002' then 3
                                when pkproduto='001001001003' then 4
                                when pkproduto='001001001005' then 5
                                when pkproduto='001002001004' then 7
                                when pkproduto='001002003001' then 8
                                when pkproduto='001003001001' then 9 end);
```

-- No caso de erro em uma das tabelas com a Chave Estrangeira de PRODUTOS escreva aqui o comando de correção dos valores.

2. O seguinte comando deve ser executado na base de dados:

```
INSERT INTO feitospor
VALUES
(1,10,5, 'Ajuste do parachoque', 2,'2019-02-05'),
(2,10,10,'Ajuste do parabrisa', 2,'2019-02-05'),
(3,10,15,'Troca da Porta Direita',2,'2019-02-05'),
(4,10,20,'Pintura dos ajustes', 2,'2019-02-05'),
(5,15,5, 'Ajuste do parachoque', 2,'2019-02-05'),
(6,15,10,'Ajuste do parabrisa', 2,'2019-02-05'),
(7,20,15,'Troca da Porta Direita',2,'2019-02-05'),
(8,20,20,'Pintura dos ajustes', 2,'2019-02-05');
```

3. Escreva um comando que atualize todos os registros das notas fiscais de venda segundo a regra: o valor unitário deve ser 1.75 se o produto for 1, 2.45 se o produto for 2, 2.95 se o produto for 3, 3.66 se o produto for 4 e 2.50 se o produto for 5 e se o produto não for nenhum valor entre 1 e 5 então o valor unitário deve ser 1.90.

4. Um analista de sistemas trabalhava na administração de um banco de dados e recebeu a ordem de alterar os valores unitários dos registros de itens de notas fiscais. Depois de executar o comando ele precisa atualizar o valor total das notas de vendas. Escreva um comando que atualize a tabela das notas de vendas calculando o valor total das notas como sendo a somatória dos valores unitários multiplicados pelas respectivas quantidades em cada item de nota correspondente a uma nota fiscal.

5. Os professores que têm residência em logradouros cujos valores de códigos estão entre 150 e 250 devem estar habilitados a ministrar aulas em cursos com carga horária maior ou igual a 100 horas. A partir do dia 01 de novembro de 2018 (sendo que se considere esta data como data de habilitação, de capacitação e de cadastramento na tabela habilitações). As combinações possíveis para esta condição podem ser geradas a partir do produto cartesiano entre professores e cursos. Escreva o comando que gere o conjunto de dados (linhas) na tabela habilitações supondo que em habilitações o valor máximo do campo id seja 28 e deva ser incrementado de 1 em 1.

6. Escreva um comando que exiba uma lista com os departamentos e seus departamentos subordinados em primeiro nível (somente o departamento principal e seu subdepartamento, sem mostrar subdepartamento de outros subordinados). Esta lista orienta a destinação de verba de investimento no desenvolvimento dos projetos destes departamentos.

7. Exiba uma lista com os nomes dos funcionarios de cada departamento e para cada departamento em ordem por funções desempenhadas pelos funcionarios. Esta lista deve ter os dados dos departamentos (exiba Código e nome do departamento), onde o Código do departamento esteja entre A01 e D01, também devem ser exibidos o nome da função (ou cargo) ao lado do Código da função:

8. Um executivo quer receber um relatório onde se vejam os funcionários contratados antes do ano 2000 ou que tenham o nível de escolaridade acima ou igual ao Ensino Superior, estes funcionários também devem ter uma renda anual superior à 40000. Os dados devem ser ordenados pelo nome de departamento ao qual o funcionário pertence:

9. Escreva os comandos que criem três tabelas (não temporárias). A primeira com nome func1 com os dados da tabela funcionários que pertençam aos departamentos com nome 'Pesquisa de ferramentas de arquitetura computacional' ou 'Desenvolvimento de Software'. A segunda tabela deve ter o nome de func2 e deve ter os dados de funcionários que foram contratados antes do ano 2000. A terceira tabela deve ter o nome de dept e deve conter os dados de departamento onde o nome do departamento seja 'Oceanografia aplicada em des. de software' ou 'Desenvolvimento de Software'

10. Escreva um comando que exiba os dados dos funcionários, os nomes de suas funções e os nomes dos departamentos onde trabalham de todos os funcionários que trabalhem em departamentos onde o departamento superior seja o 'Pesquisa em Biociências'.

11. Escreva um comando que exiba uma lista com os códigos e nomes dos departamentos com os dados agrupados e ordenados por departamento e sexo e mostre os valores do salário anual mínimo, médio e máximo, bem como a quantidade de funcionários e o salário total de cada departamento (separados por sexo). Devem ser exibidos somente os departamentos com salário médio anual acima de 150000.

Dica: para emitir um título para cada valor do sexo de cada departamento pesquise e use o sub-comando CASE...END.

12. Um gerente de produção procura você e te faz esse pedido: "Preciso de uma lista simples com os códigos e nomes dos projetos e com as quantidades de tarefas que foram executadas nos projetos sempre nos últimos 45 dias". Escreva um comando SQL que consiga atender a este requisito do gerente de produção.

13. Escreva um comando em SQL (somente UM) que realize a divisão entre projetos, tarefasprojetos e tarefas e que exiba uma lista com os códigos de tarefas que foram realizadas em todos os projetos. Para testar (e obter sucesso) no seu comando, você poderá atualizar as tabelas envolvidas nesta pergunta. SE você fizer isso, escreva no espaço complementar os comandos que você executou para atualizar as tabelas do banco de dados.

Espaço complementar para os eventuais comandos de atualização das tabelas do banco de dados.

14. Escreva um comando em SQL usando o operador JOIN que seja equivalente a este comando:

```
select * from autores, autorias, livros, bibliografia, disciplinas
      where autores.id=autorias.autoresid and
            autorias.livrosid=livros.id and
            livros.id=bibliografia.livrosid and
            bibliografia.disciplinasid=disciplinas.id;
```

15. Escreva um comando em SQL que emita uma lista com o seguinte cabeçalho: idprojeto, txnomeprojeto, txnometarefa, txprenomes, txsobrenome.
