

- 1) O **BEGIN/END** no contexto de um objeto de linguagem procedural (**PL/pgSQL**) funcionam como delimitadores de escopo de um bloco de código dentro de uma transação iniciada anteriormente. Funcionam mais como um bloco de controle, podendo conter exceções e definição de variáveis.  
O **BEGIN** no cliente **psql** é usado para iniciar uma transação, agrupando uma série de comandos DML de forma atômica. A transação ocorre com sucesso quando COMMIT ou é desfeita usando ROLLBACK.
- 2) **Resposta: E.** O operador correto para **concatenação** de string no Postgresql é o ||. **+** é para **soma**, **dual** é um comando do **oracle** (usado para criar uma tabela “imaginária” quando você quer executar uma função que não dependa de selecionar nenhum dado ex: sysdate)
- 3) **Erradas: D e E.** A criação de um índice não garante que as consultas serão sempre mais rápidas, podendo ser mais lento do que um scan sequencial. A criação de um índice consome espaço e piora o desempenho de operações como INSERT, UPDATE e DELETE.
- 4) **Resposta: B e C.** Uma **VIEW** é criada pelo comando **CREATE VIEW**, nomes de objetos (ex: tabelas, views, funções etc) devem ser únicos e uma view persiste no catálogo de dados, não sendo eliminadas quando o servidor para.
- 5)

```
SELECT ID, FIRST_NAME, MANAGER_ID, SALARY
FROM EMPREGADOS
WHERE SALARY > (
    SELECT MAX(SALARY)
    FROM EMPREGADOS
    WHERE MANAGER_ID = 100
)
ORDER BY SALARY
```

- 6) A. Verdadeiro; B. Falso; C. Falso; D. Falso; E. Verdadeiro
- 7) **Resposta C:** quando criamos um database usando o comando CREATE DATABASE <db\_name> o dono torna-se quem executou o comando.