

Programação de Banco de Dados com SQL

20-1: Garantindo Resultados de Qualidade em Consultas - Técnica Avançada

Atividades Práticas

Objetivos

- Criar uma consulta para produzir dados específicos
- Modificar uma consulta para produzir os dados especificados

Tente/solucione

1. Crie outras tabelas usadas nesta seção executando as seguintes instruções:

```
CREATE TABLE emp AS SELECT * FROM employees;  
CREATE TABLE dept AS SELECT * FROM departments;
```

2. Produza um relatório que liste o nome, o tipo, bem como o nome e a posição das colunas de todas as constraints da tabela JOB_HISTORY, além das constraints não nulas.
3. Crie uma constraint de chave primária na coluna employee_id da tabela emp
4. Crie uma chave primária na coluna department_id da tabela dept
5. Adicione uma constraint de chave estrangeira entre DEPT e EMP de modo que somente departamentos válidos possam ser inseridos na tabela EMP. Verifique se é possível excluir uma linha da tabela DEPT e se as linhas referenciadas na tabela EMP são excluídas.
6. Teste a constraint de chave estrangeira que você acabou de criar:

Conte o número de linhas na tabela EMP.
Remova o departamento 10 da tabela dept.
Agora conte o número de funcionários novamente. Deverá haver menos funcionários.

7. Produza um relatório que retorne o sobrenome, o salário, o número do departamento e a média salarial de todos os departamentos em que o salário é superior à média.
8. Crie uma view chamada V2 que retorne o salário mais alto, o salário mais baixo, a média salarial e o nome do departamento.
9. Crie uma view chamada Dept_Managers_view que retorne uma lista dos nomes de departamentos, com a inicial e o sobrenome do gerente de cada departamento. Teste a view retornando todas as linhas contidas nela. Certifique-se de que nenhuma linha possa ser atualizada por meio da view. Tente executar uma instrução UPDATE na view.
10. Crie uma sequência chamada ct_seq com todos os valores padrão.
11. Examine a instrução INSERT a seguir e corrija os erros.

```
INSERT INTO emp
(employee_id, first_name, last_name, email, phone_number, hire_date,
 job_id, salary, commission_pct, manager_id, department_id)
VALUES
(ct_seq.nextvalue, "Kaare", 'Hansen', 'KHANSEN', '44965 832123',
 sysdate, 'SA_REP', $6500, null, 100, 20);
```

12. Crie uma instrução SQL para listar todas as tabelas de usuário que contêm o nome PRIV.
13. Conceda acesso de seleção a PUBLIC na tabela EMP e verifique se o acesso foi concedido executando esta consulta.

```
SELECT *
FROM user_tab_privs
WHERE table_name = 'EMP';
```

14. Substitua ?? na consulta a seguir usando expressões regulares para retornar somente os número desta string: 'Oracle Academy9547d6905%&^ db apex'.

```
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex', '??', '')
  regexreplace
FROM DUAL;
```

15. Modifique a consulta anterior usando expressões regulares para retornar o número de dígitos desta string: 'Oracle Academy9547d6905%&^ db'

```
SELECT LENGTH(REGEXP_REPLACE('Oracle Academy9547d6905%&^ db  
apex','??','')) regexpreplace  
FROM DUAL;
```

16. Modifique a consulta novamente para retornar somente os caracteres não numéricos.

```
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','??','')  
regexpreplace  
FROM DUAL;
```

17. Usando as junções proprietárias Oracle, crie uma instrução que retorne todos os employee_ids unidos a todos os department_names.

18. Ainda usando Junções Oracle, corrija a instrução anterior para que retorne somente o nome do departamento no qual o funcionário realmente trabalha.

19. Ainda usando Junções Oracle, crie uma consulta que liste o sobrenome, o nome do departamento, o salário e o nome do país de todos os funcionários.

20. Ainda usando a sintaxe da junção Oracle, altere a consulta anterior para que também inclua o registro do funcionário sem department_id, 'Grant'.