

ATIVIDADE 1 - VIEWS - ANA BEATRIZ ROCHA DUARTE

1.

Definição	Vocabulário Correspondente
Um subconjunto de dados de uma ou mais tabelas gerados a partir de uma consulta e armazenados como uma tabela virtual	View
Nome da view	VIEW_NAME
Cria uma view, independentemente de as tabelas de base existirem ou não	Opção FORCE
Deriva dados de uma tabela, não contém funções ou grupos e pode executar operações DML por meio da view	View Simples
Criará a view somente se a tabela de base existir	Opção NOFORCE
Instrução usada para criar uma nova view	CREATE VIEW
Especifica um nome para cada expressão selecionada pela consulta da view	Alias
Uma instrução SELECT completa	Subconsulta
Deriva dados de uma ou mais tabelas, contém funções ou grupos de dados e nem sempre permite operações DML por meio da view	View Complexa

Criará novamente a view se ela já existir	Opção OR REPLACE
---	-------------------------

1. Quais são os três usos de uma view da perspectiva de um DBA?

Da perspectiva de um Administrador de Banco de Dados (DBA), os três principais usos de uma view são:

1. **Segurança e Controle de Acesso:** Views permitem restringir o acesso dos usuários a dados específicos. É possível ocultar colunas (como salários, dados pessoais) ou linhas, mostrando apenas as informações que são relevantes e permitidas para um determinado usuário ou grupo.
2. **Simplificação de Consultas Complexas:** Uma view pode encapsular uma consulta complexa que envolve múltiplas junções (JOINS), funções e cálculos. Em vez de escrever a consulta complexa repetidamente, os usuários podem simplesmente fazer um SELECT na view, tornando o acesso aos dados mais fácil e menos propenso a erros.
3. **Independência Lógica de Dados:** As views proporcionam uma camada de abstração entre os usuários e as tabelas físicas. Se a estrutura das tabelas de base for alterada (por exemplo, uma tabela é dividida em duas), o DBA pode modificar a view para que ela continue a apresentar os dados da mesma forma para o usuário final, sem que a aplicação precise ser reescrita.

2. Crie uma view simples chamada view_d_songs que contenha o ID, o título e o artista da tabela da DJs on Demand para cada código do tipo “New Age”. Na subconsulta, use o alias “Song Title” para o título da coluna.

```
CREATE VIEW view_d_songs AS
SELECT id, title AS "Song Title", artist
FROM d_songs
WHERE type_code = 'New Age';
```

3. SELECT * FROM view_d_songs. Qual foi o resultado retornado?

O resultado seria uma tabela virtual contendo três colunas: id, Song Title e artist. Ela exibiria todas as linhas da tabela d_songs em que o valor da coluna type_code é 'New Age'.

4. REPLACE view_d_songs. Adicione type_code à lista de colunas. Use aliases para todas as colunas.

```
CREATE OR REPLACE VIEW view_d_songs (ID_DA_MUSICA, TITULO_DA_MUSICA, ARTISTA, CODIGO_DO_TIPO) AS
SELECT id, title, artist, type_code
FROM d_songs
WHERE type_code = 'New Age';
```

5. Jason Tsang, o DJ da DJs on Demand, precisa de uma lista dos eventos passados e dos planejados para os próximos meses a fim de que possa tomar as providências necessárias para a preparação do equipamento de cada evento. Como gerente da empresa, você não quer que ele tenha acesso ao preço que os clientes pagaram pelos eventos. Crie uma view para ser usada por Jason que exiba o nome e a data do evento e a descrição do tema. Use aliases para cada nome de coluna.

```
CREATE VIEW view_agenda_eventos_dj AS
SELECT
    event_name AS "Nome do Evento",
    event_date AS "Data do Evento",
    theme_description AS "Descrição do Tema"
FROM d_events;
```

6. Segundo a política da empresa, somente a alta gerência tem permissão de acessar os salários de cada funcionário. Entretanto, os gerentes dos departamentos precisam saber os salários mínimo, máximo e a média salarial, agrupados por departamento. Use o banco de dados Oracle a fim de preparar uma view que exiba as informações necessárias para esses gerentes.

```
CREATE VIEW view_salarios_por_departamento AS
SELECT
    d.department_name AS "Departamento",
    MIN(e.salary) AS "Salário Mínimo",
    MAX(e.salary) AS "Salário Máximo",
    ROUND(AVG(e.salary), 2) AS "Média Salarial"
FROM employees e
JOIN departments d ON e.department_id = d.department_id
GROUP BY d.department_name;
```