SISTEMAS DE BANCO DE DADOS 2

AULA 7

VIEW

Visão em Banco de Dados

Vandor Roberto Vilardi Rissoli

APRESENTAÇÃO

- Visão em Banco de Dados (*view*)
- Características da View
- Materização em Oracle
- Referências

VISÃO EM BANCO DE DADOS

VISÃO (VIEW)

Uma **visão**, mas conhecida como **view** na área de BD, corresponde a um conjunto de tuplas resultantes de uma consulta armazenada sobre uma ou mais tabelas do SGBD.

Os usuários do SGBD podem utilizar a **view** para consultar seus conjuntos de dados como fariam sobre tabelas convencionais que armazenam dados fisicamente no SGBD.

VISÃO EM BANCO DE DADOS

Tabela

Matricula	Nome	CodCargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	08	12/01/96	21
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31
279	Carla	1	Programador	01	17/07/96	27
279	Carla	1	Programador	08	12/01/96	20
279	Carla	1	Programador	12	21/03/96	51
301	Ana	2	Analista	08	21/03/96	16
306	Manoel	1	Programador	01	21/03/96	67

Visão

Matricula	NomeCargo	CodProj
120	Programador	01
121	Programador	01
273	Projetista	01
279	Programador	01

VISÃO EM BANCO DE DADOS

- Podem ser apresentadas combinações ou subconjuntos lógicos de dados por meio da criação de **visões** interessantes sobre as tabelas existentes no SGBD;
- Uma **view** é uma "**tabela lógica**" baseada em uma ou mais tabelas reais existentes no SGBD, ou mesmo sobre outra **view**;
- A **view** em si NÃO contém dados, mas é semelhante a uma JANELA, por meio da qual é possível exibir, e até alterar, alguns dados provenientes de tabelas que persistem do SGBD.

VISÃO EM BANCO DE DADOS

PROPRIEDADES DA VIEW

- A tabela (ou tabelas) na qual uma **view** é baseada é denominada **TABELA BASE**;
- A **view** é armazenada como uma instrução **SELECT** no **Dicionário de Dados**;
- Diferentemente de tabelas reais, as **visões** NÃO são objetos físicos do SGBD e por isso não ocupam espaço em disco;
- Também podem ser definidas como um objeto que NÃO armazena dados, pois não é uma tabela, sendo composta dinamicamente por uma consulta que é previamente analisada e otimizada pelo SGBD.

VISÃO EM BANCO DE DADOS

INSTRUÇÃO SQL DE VIEW

A instrução geral SQL (DDL) de criação de uma **view**:

```
CREATE VIEW CANDIDATOS_2004 (
   registro, candidato, cargo, cidade, estado) AS
      SELECT  p.codigo, p.nome, l.cargo, l.cidade,
              l.sigla
        FROM  PESSOA p, LOCAL l
       WHERE  p.idPessoa = l.idPessoa
         AND  l.ano = 2004 ;
```

VISÃO EM BANCO DE DADOS

Após sua criação confira a existência desse novo objeto na base de dados do projeto de banco de dados.

```
DESC CANDIDATOS_2004;
```

Confira a descrição do objeto que pode parecer uma tabela, mas é uma visão (**view**).

```
SELECT  *
    FROM  CANDIDATOS_2004;
```

CARACTERÍSTICAS DA VIEW

- As alterações nos dados da tabela base da **view**, consequentemente, alteram os resultados gerados pelas consultas armazenadas na **view**;
- A criação de uma **view** é uma instrução DDL;
- O uso de **view** simplifica a interação entre usuário final e o banco de dados;
- A **view** pode ser usada como mecanismo de segurança, restringindo o acesso dos usuários;
- Em alguns bancos de dados com tecnologia **No-SQL**, as visões são a única maneira de consultar dados.

CARACTERÍSTICAS DA VIEW

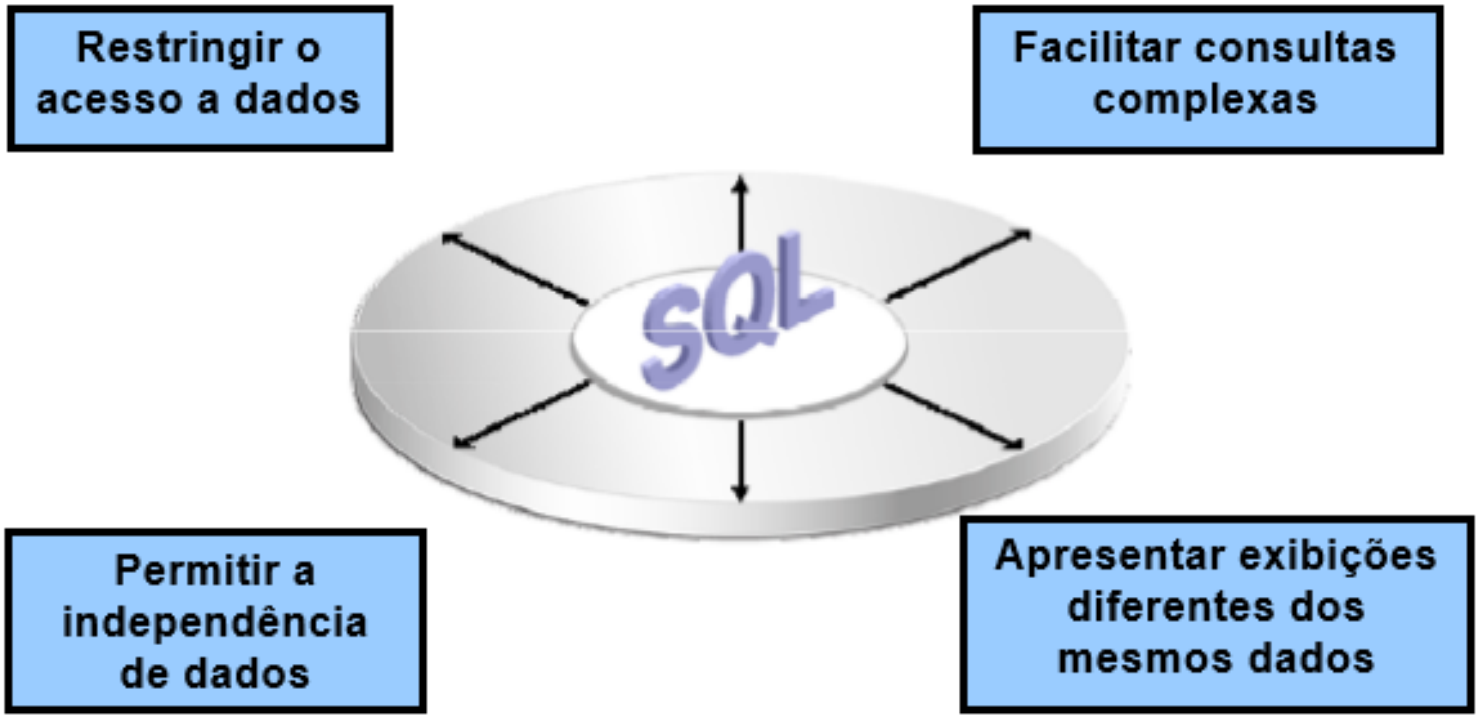
PRINCIPAL UTILIDADE

1. Aumento da **segurança** por propiciar uma visão limitada e controlada dos dados que podem ser obtidos de uma base de dados por seus usuários. Essa segurança irá depender do SGBD utilizado;

2. A **performance** do SGBD que utilizará uma consulta previamente elaborada e otimizada, não sendo necessário o processo de otimização dessa consulta quando for realizada.

CARACTERÍSTICAS DA VIEW

Algumas aplicações interessantes para este novo objeto de banco de dados (**view**).



CARACTERÍSTICAS DA VIEW

VIEW SIMPLES E COMPLEXAS

- As visões podem ser classificadas em Simples ou Complexas;
- A diferença básica está relacionada na realização das operações DML (INSERT, UPDATE e DELETE) pela **view**, manipulando as tabelas reais (base);

RECURSOS	VIEW SIMPLES	VIEW COMPLEXA
Número de tabelas	Uma	Uma ou mais
Contêm funções	Não	Sim
Contêm grupos	Não	Sim
Permitem a execução de operações DML	Sim	Nem sempre

CARACTERÍSTICAS DA VIEW

- As complexidades possíveis com a view dependem dos recursos disponíveis no SGBD utilizado;
- A diferença básica entre **Simples** e **Complexa** está relacionada a realização das operações DML (INSERT, UPDATE e DELETE) pela **view**;
  - SIMPLES:
    - Deriva dados de uma única tabela;
    - Não contém funções ou agrupamento de dados;
    - Permite a execução de operações DML.
  - COMPLEXA:
    - Deriva dados de várias tabelas;
    - Contém funções ou agrupamento de dados;
    - Nem sempre permite a execução de operações DML.

CARACTERÍSTICAS DA VIEW

- Representação de dados contidos em outras tabelas (tabelas base) ou mesmo em outras visões;
- Trata resultado de uma consulta como uma tabela
  - consulta armazenada;
  - tabela virtual;
- Espaço de armazenamento (no dicionário de dados) apenas para a consulta (SELECT) que define a **view**;
- A consulta é executada cada vez que a visão é acessada.

CARACTERÍSTICAS DA VIEW

- **Utilidade**:
  - SEGURANÇA - restrição de acesso as tuplas e as colunas;
  - ARMAZENAMENTO de consultas complexas ou executadas com muita frequência
    - Simplicidade para o usuário;
    - Abstração;
  - APRESENTAÇÃO dos dados com menor complexidade ou em diferentes perspectivas;
  - ISOLAMENTO de aplicações em relação a alterações de esquema;

CARACTERÍSTICAS DA VIEW

SQL DE VIEW PARA DIFERENTES SGBD

Observe as instruções SQL comparando a sintaxe entre os diferentes SGBDs.

MySQL

```
CREATE [OR REPLACE]
  [ALGORITHM = {UNDEFINED|MERGE|TEMPTABLE}]
  [DEFINER = { user | CURRENT_USER }]
  [SQL SECURITY { DEFINER | INVOKER }]
  VIEW nome da view [(colunas)]
   AS expressão SELECT
   [WITH [CASCADED | LOCAL] CHECK OPTION]
```

CARACTERÍSTICAS DA VIEW

ORACLE

CREATE [OR REPLACE]

VIEW nome da view

[(*NomeColuna* [, *NomeColuna* ...])]

AS expressão SELECT

[WITH CHECK OPTION | READ ONLY] ;

CARACTERÍSTICAS DA VIEW

- **Operações sobre visões** **ORACLE**
  - Visões não atualizáveis (***read-only***)
    - Seleção;
  - Visões atualizáveis (***updatable***)
    - Seleção, inserção, remoção, atualização;
- **Privilégios**
  - Proprietário da view *(owner*):
    - Operações requerem privilégios adequados sobre a tabela base;
    - Pode conceder privilégios de acesso
      - se for dono das tabelas base;
      - se tiver recebido os privilégios com *grant option*
  - Outros usuários: requerem privilégios para *view;*

CARACTERÍSTICAS DA VIEW

ORACLE

- Visões **inerentemente** **NÃO** atualizáveis, contêm:
  - Operadores de conjunto (UNION, INTERSECT, MINUS,...);
  - Operador DISTINCT
  - GROUP BY (como parte da visão);
  - ORDER BY;
  - Subconsulta na lista da cláusula SELECT;
  - *stored procedures*
  - Alguns casos de junções.

CARACTERÍSTICAS DA VIEW

- Suponha a tabela **DISCIPLINA**:

  {sigla, nome, nCred, professor, livro}

- Exemplo simples com todas as tuplas de uma única tabela (visão atualizável)

```
CREATE OR REPLACE VIEW
    V_DISCIPLINA AS SELECT nome, sigla
                        FROM DISCIPLINA;
```

- **Testando a criação da view com uma instrução DML de consulta**

```
SELECT * FROM V_DISCIPLINA;
```

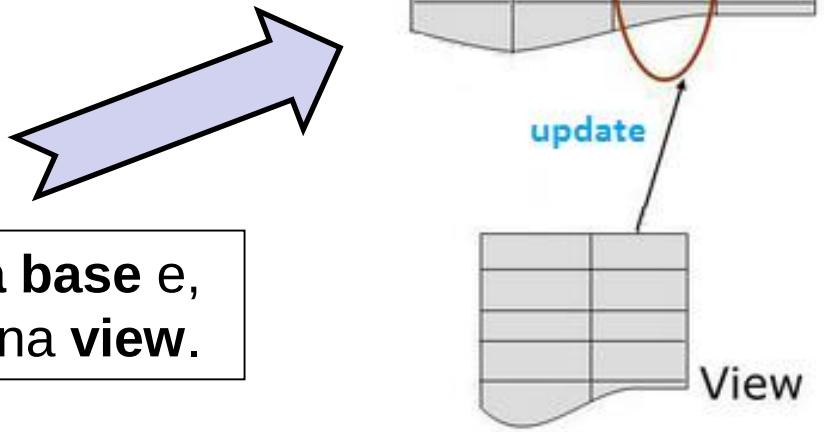
- **Realizando alteração (DML) pela view**

```
UPDATE V_DISCIPLINA SET nome = 'Alg1.1'
  WHERE sigla = 'SCC181';
```

CARACTERÍSTICAS DA VIEW

- **DISCIPLINA** {sigla, nome, nCred, professor, livro}
- **Alteração sobre a view**

```
UPDATE V_DISCIPLINA SET nome = 'Alg1.1'
  WHERE sigla = 'SCC181';
```



A atualização ocorre na **tabela base** e, consequentemente, se reflete na **view**.

CARACTERÍSTICAS DA VIEW

ORACLE

- Opção para tornar a *view* *READ ONLY*
  (só de leitura)

```
CREATE VIEW v_disciplina AS
    SELECT nome, sigla
      FROM  DISCIPLINA
WITH READ ONLY;
```

CARACTERÍSTICAS DA VIEW

- Suponha a tabela **PROFESSOR**:

  {nome, nFunc, idade, titulacao}

- Exemplo de manipulação simples pela view

```
CREATE OR REPLACE VIEW
   V_PROFESSOR_DOUTOR AS
     SELECT * FROM PROFESSOR
     WHERE  titulação = 'DOUTOR' ;
```

- Escrevendo na tabela base pela view

```
INSERT INTO V_PROFESSOR_DOUTOR VALUES
 ('Roberto',43, 45, 'TITULAR' );
```

→ Operação DML realizada com sucesso.

CARACTERÍSTICAS DA VIEW

- Analisando as operações a seguir sobre este domínio de dados sobre o professor averigue qual das instruções estariam adequadas para recuperar e mostrar a inserção da tuplas do **Roberto**?

  a) **SELECT * FROM PROFESSOR;**

  b) **SELECT * FROM V_PROFESSOR_DOUTOR;**

→ A consulta **a** mostraria a inserção realizada pela view, pois a **b** só coleciona as tuplas com titulação de doutor.

CARACTERÍSTICAS DA VIEW

- A outra opção na instrução de criação da visão é comum a todos os SGBDs e restringe a inserção de tuplas que não atendam as definições da visão

WITH CHECK OPTION

Suponha a mesma visão, mas com está opção:

```
CREATE OR REPLACE VIEW
    V_PROFESSOR_DOUTOR AS
        SELECT * FROM PROFESSOR
        WHERE  titulação = 'DOUTOR'
WITH CHECK OPTION;
```

CARACTERÍSTICAS DA VIEW

- Escrevendo na tabela base pela view

  INSERT INTO V_PROFESSOR_DOUTOR VALUES
  ('Roberto',43, 45, 'TITULAR');

  → Operação DML de inserção NÃO será realizada devido a criação da view com WITH CHECK OPTION que só aceitará **DOUTOR**.

- **A opção WITH CHECK OPTION deverá ser usada em** visões atualizáveis, que não permitirão a execução de operações que violem a condição de seleção que define a visão.

CARACTERÍSTICAS DA VIEW

VIEW DE JUNÇÃO DE TABELAS

Uma outra alternativa para criação de uma visão é a junção de tabelas para formar a view conveniente.

- Exemplo de uma visão a partir de uma JUNÇÃO
- Suponha as tabelas a seguir:

  ALUNO = {nome, matricula, idade, DtNasc}

  DISCIPLINA = {sigla, nome, nCred, professor, livro}

  MATRICULA = {sigla, numero, aluno, ano, nota}

CARACTERÍSTICAS DA VIEW

- Exemplo de uma visão a partir de uma JUNÇÃO
- Respeitando as tabelas descritas anteriormente será criada uma nova visão:

```
CREATE VIEW V_MATRICULA
   (matricula, nome, sigla, disciplina) AS
     SELECT a.matricula, a.nome, d.sigla, d.nome
       FROM ALUNO a JOIN MATRICULA m
                      ON a.matricula = m.aluno
                         JOIN DISCIPLINA d
                          ON m.sigla = d.sigla;
```

CARACTERÍSTICAS DA VIEW

VIEW DE JUNÇÃO ATUALIZÁVEL

Um recurso interessante para alguns SGBDs é a junção para visões atualizáveis.

- *VIEW de Junção ATUALIZÁVEL (Updatable join views)*
  - Regra geral:
    - Operações DML podem modificar **apenas 1** das tabelas base por vez;
  - Conceito fundamental:
    - Existe a **Preservação de Chave** em ao menos uma das tabelas da junção.

CARACTERÍSTICAS DA VIEW

PRESERVAÇÃO DE CHAVE

- Se toda chave na tabela base é única (chave candidata) no resultado da junção;
- Ela depende da semântica e não da instância atual da view;
- Para que uma view seja atualizável, é suficiente que pelo menos uma tabela subjacente tenha preservação de chave (esta única tabela poderá ser atualizada);
- No entanto, mesmo que todas tenham preservação de chave, a atualização só pode ocorrer em uma delas por vez.

CARACTERÍSTICAS DA VIEW

- Analise as instruções SQL e identifique qual a tabela com Preservação de Chave?
- Suponha as tabelas reais ALUNO e MATRICULA:

  ALUNO = {nome, matricula, idade, dtNasc}

  MATRICULA = {sigla, numero, aluno, ano, nota}

```
CREATE VIEW V_MATRICULA
  (nome, aluno, sigla, numero, ano) AS
   SELECT a.nome, m.aluno, m.sigla, m.numero,
                m.ano
     FROM ALUNO a JOIN MATRICULA m
                  ON a.matricula = m.aluno;
```

EXERCÍCIOS DE FIXAÇÃO

- Qual o resultado das seguintes operações para as tabelas reais ALUNO e MATRICULA:

  ALUNO = {nome, matricula, idade, dtNasc}

  MATRICULA = {sigla, numero, aluno, ano, nota}

  V_MATRICULA = {nome, aluno, sigla, numero, ano}

  (A) (A, M) (M) (M) (M)

```
INSERT INTO V_MATRICULA
    values('Ana', 111, 'SCE518', 1, 2007);
```

Não preserva a chave em ALUNO

```
INSERT INTO V_MATRICULA (Aluno, Sigla)
 values(111, 'SCE228');
```

Não pode NULL em numero e ano **PK**

```
INSERT INTO V_MATRICULA(Aluno,Sigla,Numero,Ano)
    values(111, 'SCE518', 1, 2007);
```

OK

EXERCÍCIOS DE FIXAÇÃO

- Qual o resultado das seguintes operações para as tabelas reais ALUNO e MATRICULA:

  ALUNO = {nome, matricula, idade, dtNasc}

  MATRICULA = {sigla, numero, aluno, ano, nota}

  V_MATRICULA = {nome, aluno, sigla, numero, ano}

  (A) (A, M) (M) (M) (M)

```
update v_matricula set nome = 'Joana'
                       where Aluno = 111;
```

Não preserva a chave

```
update v_matricula set sigla = 'SCC518'
                       where sigla = 'SCE518';
```

OK se disciplina na Turma (FK)

```
update v_matricula set Aluno = 111
                       where nome = 'Joana';
```

OK se disciplina na Turma (FK)

```
delete from v_matricula where Aluno = 111;
```

OK apaga 111 sem apagar da tabela Aluno

EXERCÍCIOS DE FIXAÇÃO

ALUNO = {matricula, nome, idade, DtNascimento}

MONITORIA = {aluno, disciplina}

```
CREATE OR REPLACE VIEW V_MONITORIA
(matricula, nome, disciplina) AS
      SELECT a.matricula, a.nome, m.disciplina
      FROM MONITORIA m Join ALUNO a
            ON a.matricula = m.aluno;
```

- Ambas as tabelas base são atualizáveis.
- Qual é o resultado da seguinte operação?

```
DELETE FROM V_MONITORIA WHERE matricula = 111;
```

OK apaga 111 da tabela Monitoria (atualiza 1 tabela por vez)

CARACTERÍSTICAS DA VIEW

- *Updatable join views*

  - Para **INSERT**:
    - somente pode envolver colunas provenientes de **1 tabela com preservação de chave**
    - **WITH CHECK OPTION** ⇒ não são permitidas inserções

CARACTERÍSTICAS DA VIEW

- *Updatable join views*
  - Para **UPDATE:**
    - colunas atualizáveis são aquelas provenientes de **1 tabela com preservação de chave**
    - **WITH CHECK OPTION** ⇒ atributos de junção (ON) e atributos de tabelas usadas mais do que uma vez não são atualizáveis

CARACTERÍSTICAS DA VIEW

- *Updatable join views*

  - Para **DELETE**:
    - Documentação Oracle: "somente se há exatamente uma tabela com preservação de chave"
    - Testes: remoção de registros da primeira tabela usada na definição da **view**
    - **WITH CHECK OPTION** ⇒ se a tabela base que possui preservação de chave for usada mais do que uma vez, não é possível deletar

CARACTERÍSTICAS DA VIEW

- Instruções DDL que cria essa “tabela virtual” (view) que resulta, realmente, de uma consulta coerentemente elaborada:

  CREATE VIEW ...

  ALTER VIEW ...

  DROP VIEW ...

- Se obtém parte da visão de algo que interessa e que o controle de acesso também poderá administrar aos usuários.

MATERIALIZAÇÃO EM ORACLE

MATERIZALIZAÇÃO

- Visões **armazenadas** como tabelas
  - dados provenientes de *tabelas base*
- Utilidade
  - replicação de dados
  - performance
    - *snapshot* local de dados remotos
    - armazenamento de resultados de consultas complexas e custosas
  - armazenamento de informações sumarizadas
  - **distribuição** de dados

MATERIALIZAÇÃO EM ORACLE

- Comuns em *Data Warehousing* (DW), sistemas distribuídos, computação móvel....

- Principais **desvantagens**:
  - ocupa espaço de armazenamento
  - exige *refresh* quando as tabelas base são modificadas, podendo ser constante e comprometendo então o desempenho.

MATERIALIZAÇÃO EM ORACLE

- Visões Materializadas
  - por *default* : *read-only*
- Recursos Oracle de *Advanced Replication*
  - permitem que as visões materializadas sejam atualizáveis.
- Tipos
  - Visões materializadas com **agregações**;
  - Visões materializadas apenas com **junções**;
  - Visões materializadas **aninhadas**.

MATERIALIZAÇÃO EM ORACLE

Exemplo:

```
SELECT d.sigla, count(m.sigla) as
                          Nro_Matriculados
  FROM disciplina d, matricula m
 WHERE d.sigla = m.sigla
 GROUP BY d.sigla;
```

Aciona LOGS nas tabelas base para o ***refresh fast***

```
   (precisa ser criado antes da visão)
CREATE MATERIALIZED VIEW LOG ON
  disciplina WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON
  matricula WITH ROWID;
```

MATERIALIZAÇÃO EM ORACLE

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}

```
CREATE MATERIALIZED VIEW
  view_matriculados
   BUILD IMMEDIATE
   REFRESH FAST ON COMMIT
   AS SELECT d.sigla, count(m.sigla) as
                              Nro_Matriculados
        FROM disciplina d, matricula m
        WHERE d.sigla = m.sigla
        GROUP BY d.sigla;
```

MATERIALIZAÇÃO EM ORACLE

DICAS (em Oracle)

- Para consultar informações do dicionário de dados, ou seja, consultar *views* do dicionário:
  - Tabelas
    - **SELECT * FROM user_tables**
  - Visões, atributos e colunas atualizáveis
    - **SELECT * FROM user_views**
    - **SELECT * FROM user_updatable_columns**

- Documentação sobre Dicionário de Dados em *Oracle 11g Database Reference*

Onde consultar *Views*

- Documentos em *list of books* no site da ORACLE
  - SQL *Reference*
  - *Administrator's Guide*
  - *Concepts*
  - *Data Warehousing Guide*
    - tipos de visões materializadas e restrições de cada tipo para *refresh*
  - *Advanced Replication*
    - uso de visões materializadas em ambientes distribuídos – replicação de dados

Referência de Criação e Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- BANCO DE DADOS 10g
  - Fundamentos de SQL I - volume II - Guia do Aluno
- ORACLE DATABASE 11g SQL
  - Domine SQL e PL/SQL no banco de dados Oracle
- RICARDO TERRA - APOSTILA ORACLE
  - Sítio virtual:
    http://pt.slideshare.net/rterrabh/2008-apostila-oracle