OracleDB: Explorando Visões, V. Materializadas, Transações e Desempenho.

Alunos: Allan Bastos, Mateus Carvalho, Lucas Padilha e Victor Probio.

Transações

Transações São unidades lógicas atômicas de trabalho que contêm um ou mais comandos SQL.

As transações no oracle obedecem as propriedades ACID.

- Atomicidade.
- Consistência.
- Isolação.
- Durabilidade.

Comandos de transações

COMMIT;

ROLLBACK;

SAVEPOINT ‘nome’;

ROLLBACK TO SAVEPOINT ‘nome’;

SET TRANSACTION NAME ‘nome’;

Comandos estruturais: CREATE, DROP, RENAME OU ALTER

USUÁRIO 1;

```
Username? (''?) sh_usuario
Password? (**********?) *******
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT * FROM sh.channels;

   CHANNEL_ID CHANNEL_DESC    CHANNEL_CLASS       CHANNEL_CLASS_ID CHANNEL_TOTAL       CHAN
NEL_TOTAL_ID
____________ _______________ _______________ __________________ _______________ _______
____________
            3 Direct Sales    Direct                           12 Channel total
           1
            9 Tele Sales      Direct                           12 Channel total
           1
            5 Catalog         Indirect                         13 Channel total
           1
            4 Internet        Indirect                         13 Channel total
           1
            2 Partners        Others                           14 Channel total
           1

SQL> INSERT INTO sh.channels VALUES ( 6, 'Subscriptions', 'Direct',   12, 'Channel total',
1 );

1 row inserted.

SQL>
```

USUÁRIO 2;

```
Username? (''?) sh_usuario
Password? (**********?) *******
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT * FROM sh.channels;

   CHANNEL_ID CHANNEL_DESC    CHANNEL_CLASS       CHANNEL_CLASS_ID CHANNEL_TOTAL       CHAN
NEL_TOTAL_ID
____________ _______________ _______________ __________________ _______________ _______
____________
            3 Direct Sales    Direct                           12 Channel total
           1
            9 Tele Sales      Direct                           12 Channel total
           1
            5 Catalog         Indirect                         13 Channel total
           1
            4 Internet        Indirect                         13 Channel total
           1
            2 Partners        Others                           14 Channel total
           1

SQL>
```

USUÁRIO 1;

```
SQL> SELECT * FROM sh.channels;

   CHANNEL_ID CHANNEL_DESC     CHANNEL_CLASS       CHANNEL_CLASS_ID CHANNEL_TOTAL       CHA
NNEL_TOTAL_ID
____________ _______________ _______________ __________________ _______________ ______
____________
           3 Direct Sales     Direct                          12 Channel total
           1
           9 Tele Sales       Direct                          12 Channel total
           1
           5 Catalog          Indirect                        13 Channel total
           1
           4 Internet         Indirect                        13 Channel total
           1
           2 Partners         Others                          14 Channel total
           1
           6 Subscriptions    Direct                          12 Channel total
           1

6 rows selected.
```

USUÁRIO 2;

```
           1
           2 Partners        Others                          14 Channel total
           1

SQL> SELECT * FROM sh.channels;

   CHANNEL_ID CHANNEL_DESC    CHANNEL_CLASS       CHANNEL_CLASS_ID CHANNEL_TOTAL
NEL_TOTAL_ID
____________ _______________ _______________ __________________ _______________
____________
           3 Direct Sales    Direct                          12 Channel total
           1
           9 Tele Sales      Direct                          12 Channel total
           1
           5 Catalog         Indirect                        13 Channel total
           1
           4 Internet        Indirect                        13 Channel total
           1
           2 Partners        Others                          14 Channel total
           1
```

USUÁRIO 1;

```
SQL> SELECT * FROM sh.channels;

   CHANNEL_ID CHANNEL_DESC     CHANNEL_CLASS       CHANNEL_CLASS_ID CHANNEL_TOTAL       CHA
NNEL_TOTAL_ID
____________ _______________ _______________ __________________ _______________ ______
____________
           3 Direct Sales     Direct                          12 Channel total
           1
           9 Tele Sales       Direct                          12 Channel total
           1
           5 Catalog          Indirect                        13 Channel total
           1
           4 Internet         Indirect                        13 Channel total
           1
           2 Partners         Others                          14 Channel total
           1
           6 Subscriptions    Direct                          12 Channel total
           1

6 rows selected.
```

USUÁRIO 2;

```
SQL> SELECT * FROM sh.channels;

   CHANNEL_ID CHANNEL_DESC     CHANNEL_CLASS       CHANNEL_CLASS_ID CHANNEL_TOTAL
NNEL_TOTAL_ID
____________ _______________ _______________ __________________ _______________
____________
           3 Direct Sales     Direct                          12 Channel total
           1
           9 Tele Sales       Direct                          12 Channel total
           1
           5 Catalog          Indirect                        13 Channel total
           1
           4 Internet         Indirect                        13 Channel total
           1
           2 Partners         Others                          14 Channel total
           1
           6 Subscriptions    Direct                          12 Channel total
           1

6 rows selected.
```

Visões

Visões são representações lógicas de uma ou mais tabelas.

Sintaxe:

CREATE VIEW ‘nome’ AS ...

VIEW;

```sql
CREATE OR REPLACE VIEW products_sales
    AS
    SELECT     s.prod_id,
               p.prod_name,
               SUM(s.quantity_sold) AS units_sold
    FROM       sales s INNER JOIN products p
        ON s.prod_id = p.prod_id
    GROUP BY p.prod_name, s.prod_id
    ORDER BY s.prod_id;
```

RESULTADO;

Worksheet | Query Builder

```sql
SELECT * FROM products_sales
```

Query Result

SQL | Fetched 50 rows in 0.251 seconds

	PROD_ID	PROD_NAME	UNITS_SOLD
1	13	Soccer Goal - Portable	6002
2	14	Pitching Machine and Batting Cage Combo	6010
3	15	Right-Handed Graphite Shaft Iron Set	5766
4	16	Sonic Core Graphite Racquet	6929
5	17	Soccer Goal - Official	6160
6	18	Lithium Electric Golf Caddy	9591
7	19	Cricket Bat Bag	10430
8	20	Match Used Autograph Racquet	10903
9	21	Speed Trainer Bats and Training Program	5205
10	22	MLB Official Game Baseball w/ Display Case	3441
11	23	Plastic Cricket Bat	19642
12	24	Endurance Coolcore 1/2 Zip Pullover	20948
13	25	5 Point Batting Tee	19557
14	26	Pro Style Batting Tee	15950
15	27	Bucket of 24 Synthetic Baseballs	12163

Visões materializadas

Visões materializadas são o resultado de uma query que fou ‘materializada’ como um objeto de um schema.

Sintaxe:

```
CREATE MATERIALIZED VIEW view-name
BUILD [IMMEDIATE | DEFERRED]
REFRESH [FAST | COMPLETE | FORCE ]
ON [COMMIT | DEMAND ]
[[ENABLE | DISABLE] QUERY REWRITE]
AS
SELECT ...;
```

```
BEGIN DBMS_SNAPSHOT.REFRESH( '"SCHEMA"."MATERIALIZED_VIEW"','F'); end;
```

VIEW MATERIALIZADA;

```
CREATE MATERIALIZED VIEW products_sales_mv
    AS
    SELECT    s.prod_id,
              p.prod_name,
              SUM(s.quantity_sold) AS units_sold
    FROM      sales s INNER JOIN products p
        ON s.prod_id = p.prod_id
    GROUP BY p.prod_name, s.prod_id
    ORDER BY s.prod_id;
```

RESULTADO;

Worksheet | Query Builder

```
SELECT * FROM products_sales_mv;
```

Query Result

SQL | Fetched 50 rows in 0.018 seconds

	PROD_ID	PROD_NAME	UNITS_SOLD
1	13	Soccer Goal - Portable	6002
2	14	Pitching Machine and Batting Cage Combo	6010
3	15	Right-Handed Graphite Shaft Iron Set	5766
4	16	Sonic Core Graphite Racquet	6929
5	17	Soccer Goal - Official	6160
6	18	Lithium Electric Golf Caddy	9591
7	19	Cricket Bat Bag	10430
8	20	Match Used Autograph Racquet	10903
9	21	Speed Trainer Bats and Training Program	5205
10	22	MLB Official Game Baseball w/ Display Case	3441
11	23	Plastic Cricket Bat	19642
12	24	Endurance Coolcore 1/2 Zip Pullover	20948
13	25	5 Point Batting Tee	19557

Desempenho

Sitaxe e funcionamento:

EXPLAIN PLAN;

DBMS_XPLAN.DISPLAY;

SELECT * FROM TABLE (DBMS_XPLAN.DISPLAY);

CONSULTA DE DESEMPENHO;

```
EXPLAIN PLAN FOR SELECT * FROM products_sales;

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

RESULTADO;

Worksheet | Query Builder

```
EXPLAIN PLAN FOR SELECT * FROM products_sales;

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

Script Output | Query Result

SQL | All Rows Fetched: 22 in 0.344 seconds

PLAN_TABLE_OUTPUT

```
Plan hash value: 2879616603

-----------------------------------------------------------------------------------------------------------
| Id  | Operation                | Name           | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT         |                |    72 |  3816 |  3900   (3)| 00:00:01 |       |       |
|   1 |  VIEW                    | PRODUCTS_SALES |    72 |  3816 |  3900   (3)| 00:00:01 |       |       |
|   2 |   MERGE JOIN             |                |    72 |  3168 |  3900   (3)| 00:00:01 |       |       |
|   3 |    SORT JOIN             |                |    72 |  1224 |  3896   (3)| 00:00:01 |       |       |
|   4 |     VIEW                 | VW_GBC_6       |    72 |  1224 |  3896   (3)| 00:00:01 |       |       |
|   5 |      HASH GROUP BY       |                |    72 |   504 |  3896   (3)| 00:00:01 |       |       |
|   6 |       PARTITION RANGE ALL|                |  918K|  6281K|  3839   (1)| 00:00:01 |     1 |    15 |
|   7 |        TABLE ACCESS FULL | SALES          |  918K|  6281K|  3839   (1)| 00:00:01 |     1 |    15 |
|*  8 |    SORT JOIN             |                |    72 |  1944 |     4  (25)| 00:00:01 |       |       |
|   9 |     TABLE ACCESS FULL    | PRODUCTS       |    72 |  1944 |     3   (0)| 00:00:01 |       |       |
-----------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   8 - access("ITEM_1"="P"."PROD_ID")
       filter("ITEM_1"="P"."PROD_ID")
```

Utilizando índices

- B-tree

```
CREATE INDEX index_name ON table_name (column_name);
```

- Bitmap

```
CREATE INDEX index_name ON table_name (column_name);
```

- Hash

```
CREATE INDEX index_name
ON table_name (column_name)
INDEXTYPE IS HASH;
```

- Função

```
CREATE INDEX index_name ON table_name (function_name(column_name));
```

```
DROP INDEX index_name;
```

Hints

Hints são ‘sugestões’ para especificar se um índice deve ser usado ou ignorado.

Sintaxes:

INDEX(table_name index_name);

NO_INDEX(table_name index_name)

ALTER INDEX index_name UNUSABLE;

ALTER INDEX index_name REBUILD;

Consulta com índice Bitmap;

Results 1 | Results 2 | Execution plan - 1 | Execution plan - 2 | Execution plan - 3 | Execution plan - 4

Simple

Operation	Object	Optimizer	Cost	Cardinality	Bytes
SELECT STATEMENT		ALL_ROWS	1	1	151
TABLE ACCESS (BY INDEX ROWID BATCHED)	PRODUCTS	ANALYZED	1	1	151
BITMAP CONVERSION (TO ROWIDS)			0	0	0
BITMAP INDEX (SINGLE VALUE)	PRODUCTS_PROD_STATUS		0	0	0

```
SELECT * FROM PRODUCTS p WHERE PROD_STATUS = 'AVAILABLE'
```

Consulta sem índice Bitmap;

Worksheet | Query Builder

```
ALTER INDEX products_prod_status_bix UNUSABLE;

EXPLAIN PLAN FOR SELECT prod_name, prod_status FROM products WHERE prod_status = 'AVAILABLE';

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 13 in 0.032 seconds

```
PLAN_TABLE_OUTPUT
Plan hash value: 1954719464

-----------------------------------------------------------------------------
| Id  | Operation         | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |          |    36 |  1224 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| PRODUCTS |    36 |  1224 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("PROD_STATUS"='AVAILABLE')
```

README.md

Trabalho de Banco de Dados 2

Tema: Oracle DB

Conteúdos: transações, visões, visões materializadas e desempenho

Nomes dos integrantes

- Allan Bastos da Silva
  - allan.bastos@aluno.ifsp.edu.br
- Lucas dos Santos Padilha
  - mateus.lucas1@aluno.ifsp.edu.br
- Mateus Carvalho Lucas
  - lucas.padilha@aluno.ifsp.edu.br
- Victor Probio Lopes
  - victor.probio@aluno.ifsp.edu.br

Sumário

Obrigado

VictorPLopes / Projeto-BADC5-Oracle

github.com/VictorPLopes/Projeto-BADC5-Oracle