



FASE 5 – CAP 2

BRUNA LANZARINI

RM 83421

Visão Geral – Ambiente de Desenvolvimento

The screenshot displays the Oracle SQL Developer environment. The main window shows a script with the following SQL statements:

```
CREATE ROLE c##ADMINISTRADOR;  
GRANT create table, create view, create sequence, create cluster, create procedure TO c##ADMINISTRADOR;  
GRANT c##ADMINISTRADOR TO c##ADMIN_SP, c##ADMIN_RJ, c##ADMIN_DF;  
  
CREATE ROLE c##USUARIO;  
GRANT create table, create view TO c##USUARIO;  
GRANT c##USUARIO TO c##USER_SP, c##USER_RJ, c##USER_DF;
```

Below the script, there is a section for testing the script execution:

```
-----Teste  
-----SELECT * FROM all_users ORDER BY CREATED DESC;  
-----SELECT * FROM ROLE_SYS_PRIVS WHERE ROLE = 'C##ADMINISTRADOR';
```

The bottom of the window shows the results of the script execution, displaying a table with columns: ROLE, PRIVILEGE, ADMIN_OPTION, COMMON, and INHERITED. The results are as follows:

	ROLE	PRIVILEGE	ADMIN_OPTION	COMMON	INHERITED
1	C##USUARIO	CREATE TABLE	NO	NO	NO
2	C##USUARIO	CREATE VIEW	NO	NO	NO

Criação e População da Tabela

- Foi utilizado o Anexo 1 disponibilizado na área do aluno para a criação e população das 4 tabelas.



```
-- BRUNA - system
Criação_e_Populacao.sql
EXERCICIO_01.sql
EXERCICIO_02.sql
EXERCICIO_03.sql

Planilha SQL Histórico
Planilha Query Builder

-- CD_MUN NUMBER CONSTRAINT MUNICIPIOS_PK PRIMARY KEY , MUNICIPIOS
VZERAR(2(50));
-- CREATE TABLE UNIDADES_FEDERACAO
CREATE TABLE UNIDADES_FEDERACAO
( CD_UF NUMBER CONSTRAINT UNIDADES_FEDERACAO_PK PRIMARY KEY, UF
CHAR(2));
--CREATE TABLE DADOS_BRUTOS_CAPITAIS
CREATE TABLE DADOS_BRUTOS_CAPITAIS
( CD_UF NUMBER,
CD_MUN NUMBER,
ID CHAR(2),
A_2010 NUMBER,
A_2011 NUMBER,
A_2012 NUMBER,
A_2013 NUMBER,
A_2014 NUMBER,
A_2015 NUMBER,
A_2016 NUMBER,
A_2017 NUMBER,
A_2018 NUMBER,
A_2019 NUMBER,
CONSTRAINT UF_DBC_FK FOREIGN KEY (CD_UF) REFERENCES UNIDADES_FEDERACAO
(CD_UF) ENABLE,
CONSTRAINT MUNICIPIOS_DBC_FK FOREIGN KEY (CD_MUN) REFERENCES
MUNICIPIOS (CD_MUN) ENABLE,
CONSTRAINT DADOS_BRUTOS_DBC_FK FOREIGN KEY (ID) REFERENCES
DADOS_BRUTOS (ID) ENABLE
);

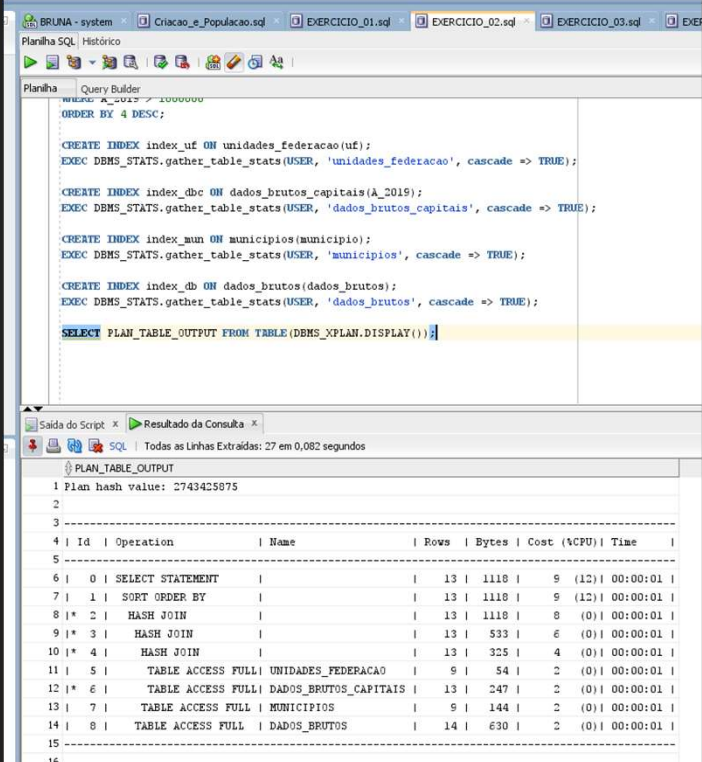
--INSERTS TABELA UNIDADES_FEDERACAO
insert into UNIDADES_FEDERACAO values ('15','PA');
insert into UNIDADES_FEDERACAO values ('23','CE');
insert into UNIDADES_FEDERACAO values ('26','PE');
insert into UNIDADES_FEDERACAO values ('29','BA');
insert into UNIDADES_FEDERACAO values ('31','MG');
insert into UNIDADES_FEDERACAO values ('33','RJ');
insert into UNIDADES_FEDERACAO values ('35','SP');
insert into UNIDADES_FEDERACAO values ('41','PR');
insert into UNIDADES_FEDERACAO values ('53','DF');

COMMIT;

--INSERTS TABELA MUNICIPIOS
insert into MUNICIPIOS values ('1501400','BELEM');
insert into MUNICIPIOS values ('2527400','SALVADOR');
insert into MUNICIPIOS values ('2304400','FORTALEZA');
insert into MUNICIPIOS values ('4106900','CURITIBA');
```


Exercício 01

- Criação de VIEW para facilitar a consulta, sem permissão de manipulação de dados (READ ONLY).



The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL script with several SQL statements for creating indexes and gathering statistics. The bottom pane shows the execution results of the script, including a plan table output.

```
CREATE INDEX index_uf ON unidades_federacao(uf);
EXEC DBMS_STATS.gather_table_stats(USER, 'unidades_federacao', cascade => TRUE);

CREATE INDEX index_dbc ON dados_brutos_capitais(A_2019);
EXEC DBMS_STATS.gather_table_stats(USER, 'dados_brutos_capitais', cascade => TRUE);

CREATE INDEX index_mun ON municipios(municipio);
EXEC DBMS_STATS.gather_table_stats(USER, 'municipios', cascade => TRUE);

CREATE INDEX index_db ON dados_brutos(dados_brutos);
EXEC DBMS_STATS.gather_table_stats(USER, 'dados_brutos', cascade => TRUE);

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

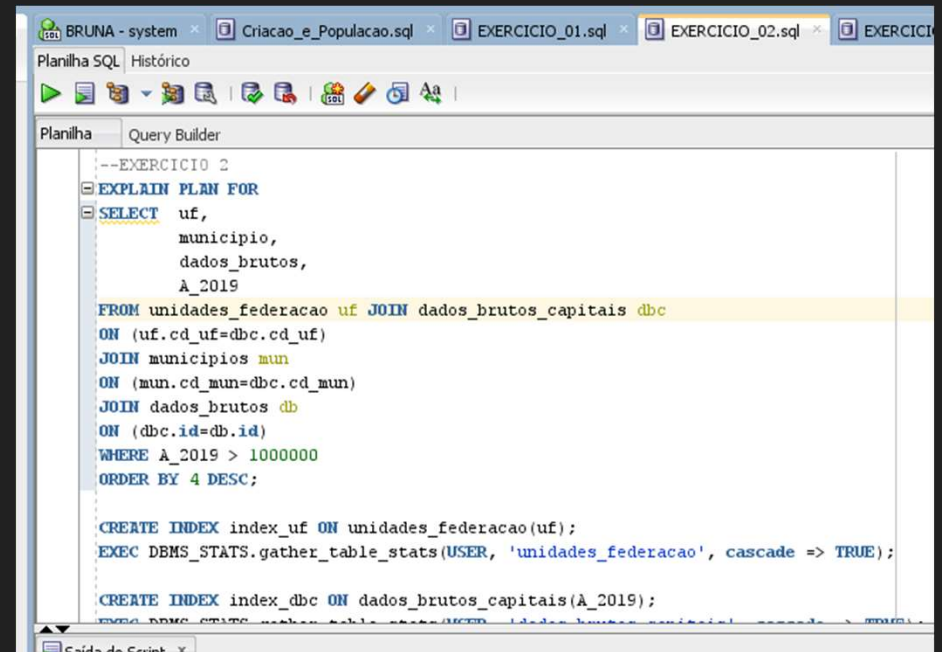
PLAN_TABLE_OUTPUT

Plan hash value: 2743425875

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		13	1118	9 (12)	00:00:01
1	SORT ORDER BY		13	1118	9 (12)	00:00:01
2	HASH JOIN		13	1118	8 (0)	00:00:01
3	HASH JOIN		13	533	6 (0)	00:00:01
4	HASH JOIN		13	325	4 (0)	00:00:01
5	TABLE ACCESS FULL	UNIDADES_FEDERACAO	9	54	2 (0)	00:00:01
6	TABLE ACCESS FULL	DADOS_BRUTOS_CAPITAIS	13	247	2 (0)	00:00:01
7	TABLE ACCESS FULL	MUNICIPIOS	9	144	2 (0)	00:00:01
8	TABLE ACCESS FULL	DADOS_BRUTOS	14	630	2 (0)	00:00:01

Exercício 02

- Criação de INDEX para otimização da query.



The screenshot shows a SQL IDE window with multiple tabs. The active tab is 'EXERCICIO_02.sql'. The window contains a SQL script with a query and index creation statements. The query is a SELECT statement with a JOIN and a WHERE clause. Below the query, there are two CREATE INDEX statements. The first index is on the 'unidades_federacao' table, and the second is on the 'dados_brutos_capitais' table. The script also includes a comment '--EXERCICIO 2' and an 'EXPLAIN PLAN FOR' statement before the SELECT query.

```
--EXERCICIO 2
EXPLAIN PLAN FOR
SELECT uf,
        municipio,
        dados_brutos,
        A_2019
FROM unidades_federacao uf JOIN dados_brutos_capitais dbc
ON (uf.cd_uf=dbc.cd_uf)
JOIN municipios mun
ON (mun.cd_mun=dbc.cd_mun)
JOIN dados_brutos db
ON (dbc.id=db.id)
WHERE A_2019 > 1000000
ORDER BY 4 DESC;

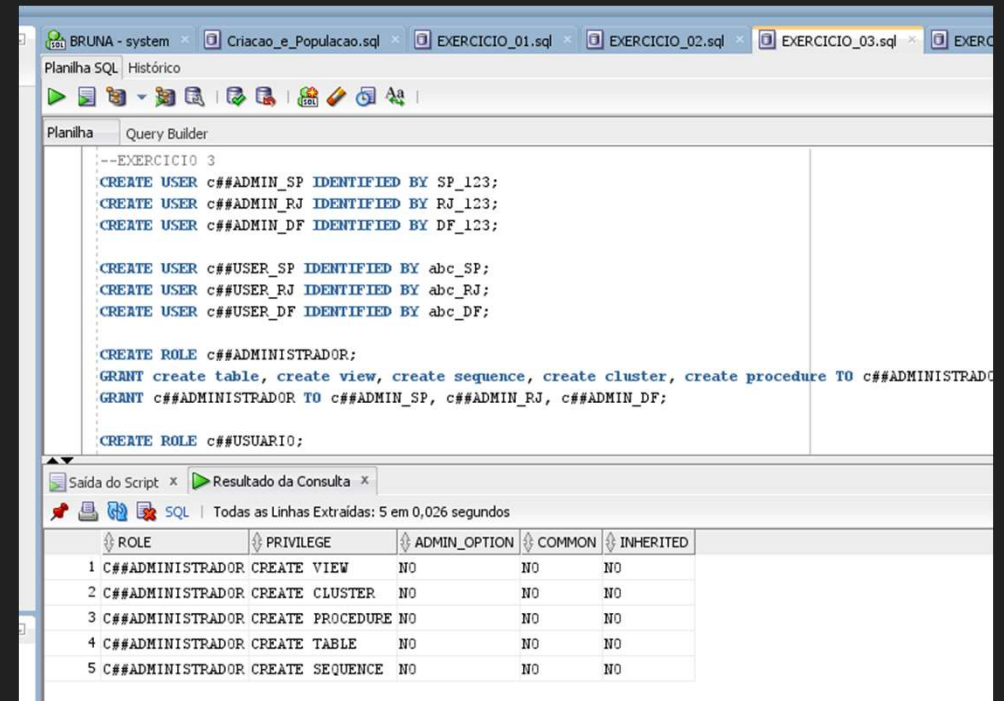
CREATE INDEX index_uf ON unidades_federacao(uf);
EXEC DBMS_STATS.gather_table_stats(USER, 'unidades_federacao', cascade => TRUE);

CREATE INDEX index_dbc ON dados_brutos_capitais(A_2019);
EXEC DBMS_STATS.gather_table_stats(USER, 'dados_brutos_capitais', cascade => TRUE);
```

Exercício 03

○ Criação de Usuários e Roles;

- Foi conversado com o professor Milton sobre a resolução de um problema com o uso de 'c##' antes das identificações, ao invés de um container local.



```
--EXERCICIO 3
CREATE USER c##ADMIN_SP IDENTIFIED BY SP_123;
CREATE USER c##ADMIN_RJ IDENTIFIED BY RJ_123;
CREATE USER c##ADMIN_DF IDENTIFIED BY DF_123;

CREATE USER c##USER_SP IDENTIFIED BY abc_SP;
CREATE USER c##USER_RJ IDENTIFIED BY abc_RJ;
CREATE USER c##USER_DF IDENTIFIED BY abc_DF;

CREATE ROLE c##ADMINISTRADOR;
GRANT create table, create view, create sequence, create cluster, create procedure TO c##ADMINISTRADOR;
GRANT c##ADMINISTRADOR TO c##ADMIN_SP, c##ADMIN_RJ, c##ADMIN_DF;

CREATE ROLE c##USUARIO;
```

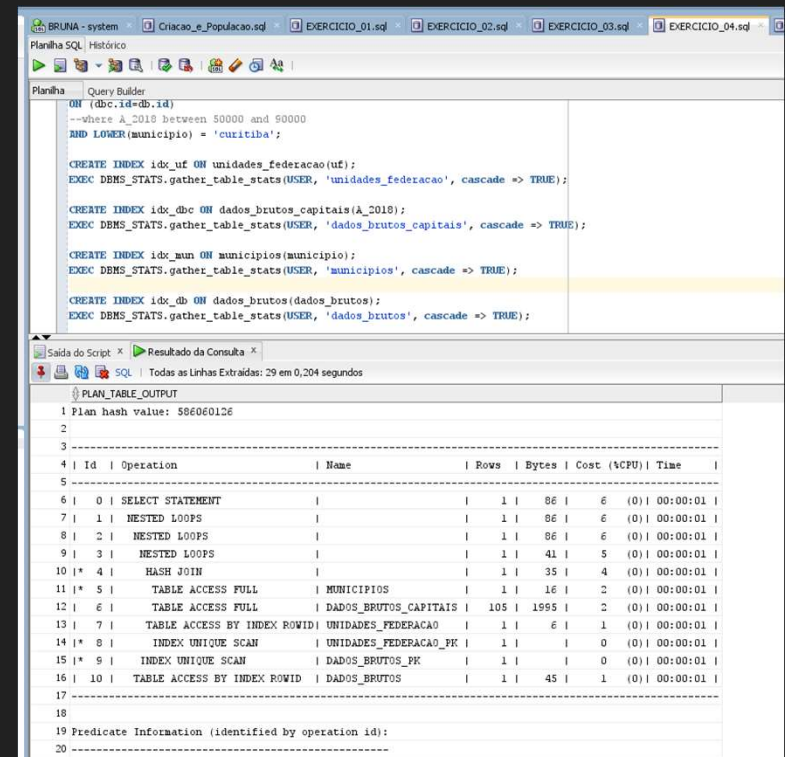
Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 5 em 0,026 segundos

ROLE	PRIVILEGE	ADMIN_OPTION	COMMON	INHERITED
1 C##ADMINISTRADOR	CREATE VIEW	NO	NO	NO
2 C##ADMINISTRADOR	CREATE CLUSTER	NO	NO	NO
3 C##ADMINISTRADOR	CREATE PROCEDURE	NO	NO	NO
4 C##ADMINISTRADOR	CREATE TABLE	NO	NO	NO
5 C##ADMINISTRADOR	CREATE SEQUENCE	NO	NO	NO

Exercício 04

- Criação de INDEX para otimização da query.



The screenshot shows a SQL IDE window with multiple tabs. The active tab is 'EXERCICIO_04.sql'. The query editor contains the following SQL code:

```
ON (dbc.id=db.14)
--where A_2018 between 50000 and 90000
AND LOWER(municipio) = 'curitiba';

CREATE INDEX idx_uf ON unidades_federacao(uf);
EXEC DEMS_STATS.gather_table_stats(USER, 'unidades_federacao', cascade => TRUE);

CREATE INDEX idx_dbc ON dados_brutos_capitais(A_2018);
EXEC DEMS_STATS.gather_table_stats(USER, 'dados_brutos_capitais', cascade => TRUE);

CREATE INDEX idx_mun ON municipios(municipio);
EXEC DEMS_STATS.gather_table_stats(USER, 'municipios', cascade => TRUE);

CREATE INDEX idx_db ON dados_brutos(dados_brutos);
EXEC DEMS_STATS.gather_table_stats(USER, 'dados_brutos', cascade => TRUE);
```

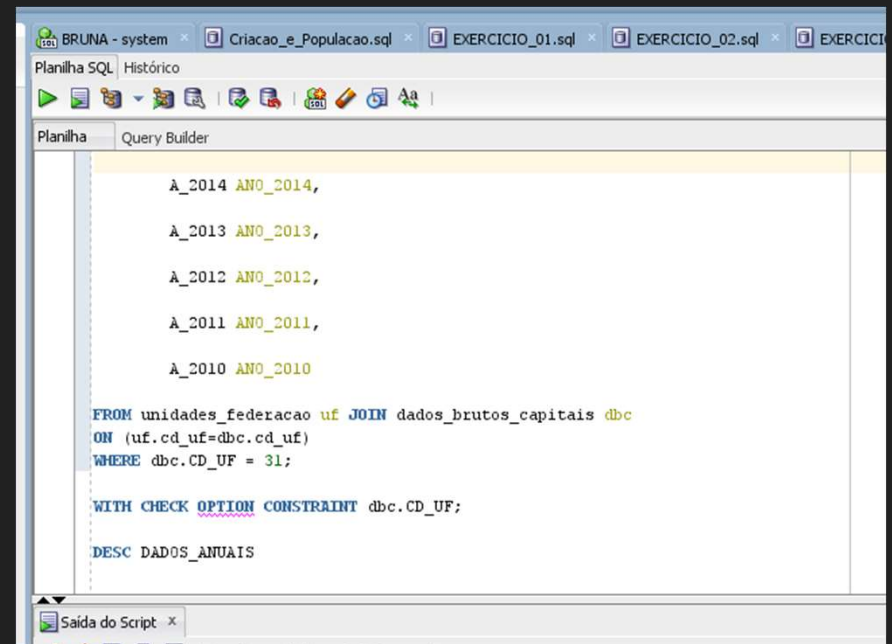
Below the query editor, the 'PLAN_TABLE_OUTPUT' section displays the execution plan for the query. The plan hash value is 586060126. The plan details are as follows:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	86	6 (0)	00:00:01
1	NESTED LOOPS		1	86	6 (0)	00:00:01
2	NESTED LOOPS		1	86	6 (0)	00:00:01
3	NESTED LOOPS		1	41	5 (0)	00:00:01
4	HASH JOIN		1	35	4 (0)	00:00:01
5	TABLE ACCESS FULL	MUNICIPIOS	1	16	2 (0)	00:00:01
6	TABLE ACCESS FULL	DADOS_BRUTOS_CAPITAIS	105	1995	2 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID	UNIDADES_FEDERACAO	1	6	1 (0)	00:00:01
8	INDEX UNIQUE SCAN	UNIDADES_FEDERACAO_PK	1	1	0 (0)	00:00:01
9	INDEX UNIQUE SCAN	DADOS_BRUTOS_PK	1	1	0 (0)	00:00:01
10	TABLE ACCESS BY INDEX ROWID	DADOS_BRUTOS	1	45	1 (0)	00:00:01

Predicate Information (identified by operation id):

Exercício 05

- Criação de uma VIEW para facilitar a consulta, sem permitir a alteração do código do município (dbc.CD_UF).



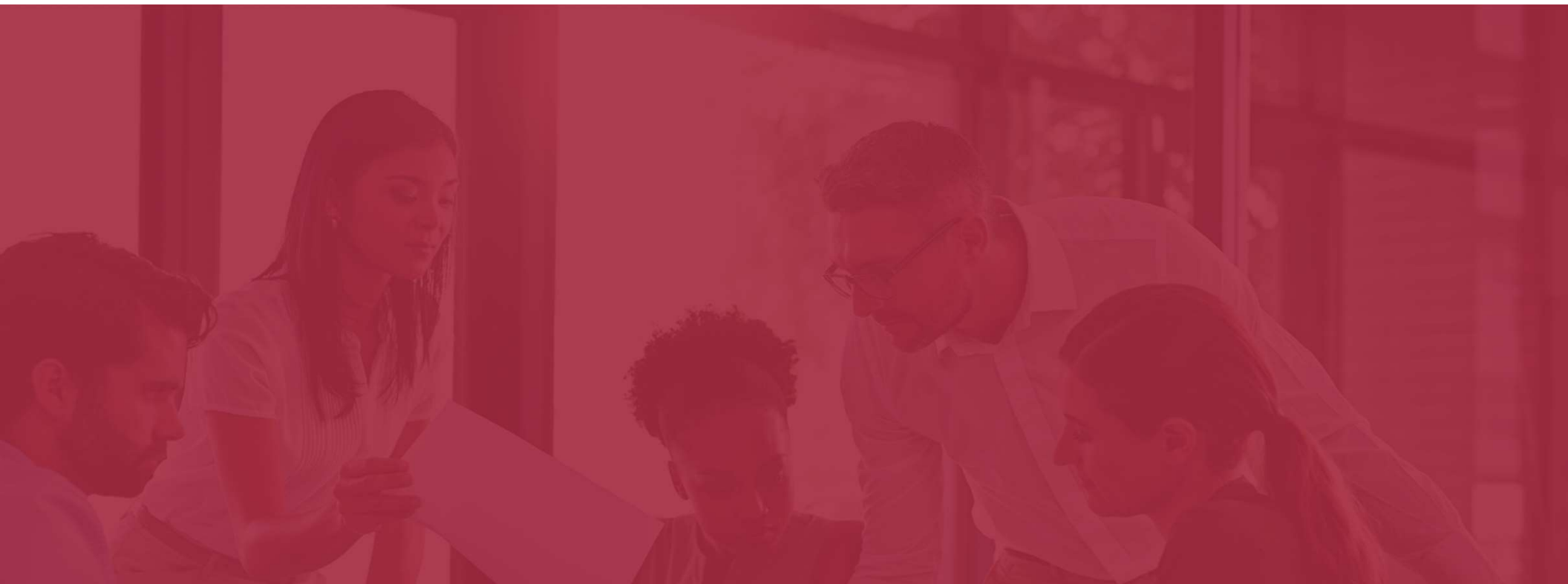
The screenshot shows a SQL IDE window with multiple tabs. The active tab is 'EXERCICIO_01.sql'. The query editor displays the following SQL code:

```
A_2014 ANO_2014,  
A_2013 ANO_2013,  
A_2012 ANO_2012,  
A_2011 ANO_2011,  
A_2010 ANO_2010  
  
FROM unidades_federacao uf JOIN dados_brutos_capitais dbc  
ON (uf.cd_uf=dbc.cd_uf)  
WHERE dbc.CD_UF = 31;  
  
WITH CHECK OPTION CONSTRAINT dbc.CD_UF;  
  
DESC DADOS_ANUAIS
```

The IDE interface includes a toolbar with various icons, a 'Planilha SQL' tab, and a 'Saída do Script' window at the bottom.

Scripts

- Você pode encontrar os scripts utilizados para a resolução de cada exercício em:
[brunadl/FASE5_CAP2_RM83421 \(github.com\)](https://github.com/brunadl/FASE5_CAP2_RM83421)



Obrigada

Bruna Lanzarini

RM 83421