

Comandos para criar banco de dados, tabelas e índices.

1 - Comando para criar um banco de dados:

```
CREATE DATABASE FUTDB2  
STOGROUP JAKASTG1  
BUFFERPOOL BPO  
CCSID EBCDIC;
```

2 - Comando para criar um table space:

```
CREATE TABLESPACE TSFUTDB2 IN FUTDB2  
BUFFERPOOL BPO  
LOCKSIZE ANY  
SEGSIZE 4  
CLOSE NO  
USING STOGROUP JAKASTG1  
PRIQTY 48  
SECQTY 48  
ERASE NO  
PCTFREE 10  
FREEPAGE 0;
```

3 - Comando para criar uma tabela DB2.

```
CREATE TABLE DBADB2.TBCADDB2  
(  
CAD_MATRIC SMALLINT NOT NULL,  
CAD_NOME CHAR(30) NOT NULL,  
CAD_CURSO CHAR(15) NOT NULL,  
CAD_TURMA CHAR(15) NOT NULL,  
CAD_END CHAR(25) NOT NULL,
```

```
CAD_BAIRRO CHAR(25) NOT NULL,  
CAD_CIDADE CHAR(30) NOT NULL,  
CAD_ESTADO CHAR(02) NOT NULL,  
CAD_TELEF CHAR(15) NOT NULL,  
CAD_EMAIL CHAR(30)  
)  
IN FUTDB22.TSFUTDB2  
;
```

4 - Comando para criar um índice e defini-lo como primary key.

```
CREATE UNIQUE INDEX  
DBADB2.ITBCADDB22 ON DBADB2.TBCADDB22  
(CAD_MATRIC ASC)  
;  
ALTER TABLE DBADB2.TBCADDB22  
PRIMARY KEY (CAD_MATRIC)  
;
```

5 - Comando para dar permissões para uso da tabela.

```
GRANT SELECT,INSERT,UPDATE,DELETE ON DBADB2.TBCADDB22 TO PUBLIC;
```

Principais comandos SQL no DB2

1 - INSERT

2 - UPDATE

3 - DELETE

4 - SELECT

5 - SELECT COM CURSOR

6- Tratamento de campo nulo

7 – JOIN de tabelas

Dados

DCLGEN da tabela DB2.

```
* DCLGEN TABLE(DBADB2.TBCADDB2) *
* LIBRARY(DBADB2.JCL.DATA(CADDB2)) *
* ACTION(REPLACE) *
* LANGUAGE(COBOL) *
* STRUCTURE(CADDB2) APOST *
```

* ... IS THE DCLGEN COMMAND THAT MADE THE FOLLOWING STATEMENTS *

```
EXEC SQL DECLARE DBADB2.TBCADDB2 TABLE
( CAD_MATRIC          SMALLINT NOT NULL,
  CAD_NOME            CHAR(30) NOT NULL,
  CAD_CURSO           CHAR(15) NOT NULL,
  CAD_TURMA           CHAR(15) NOT NULL,
  CAD_END             CHAR(25) NOT NULL,
  CAD_BAIRRO          CHAR(25) NOT NULL,
  CAD_CIDADE          CHAR(30) NOT NULL,
  CAD_ESTADO          CHAR(2) NOT NULL,
  CAD_TELEF           CHAR(15) NOT NULL,
  CAD_EMAIL           CHAR(30)
) END-EXEC.
```

01 CADDDB2.

10 CAD-MATRIC PIC S9(4) USAGE COMP.
10 CAD-NOME PIC X(30).
10 CAD-CURSO PIC X(15).
10 CAD-TURMA PIC X(15).
10 CAD-END PIC X(25).
10 CAD-BAIRRO PIC X(25).
10 CAD-CIDADE PIC X(30).
10 CAD-ESTADO PIC X(2).
10 CAD-TELEF PIC X(15).
10 CAD-EMAIL PIC X(30).

Areas necessárias para executar comando DB2.

```
EXEC SQL INCLUDE SQLCA  
END-EXEC.
```

*AREA DO DB2 EX: SQLCODE

```
EXEC SQL INCLUDE CADDDB2  
END-EXEC.
```

*AREA DOA TABELA DB2

Após cada comando é necessário testar erro.

```
IF SQLCODE = 0  
END-IF.
```

1 – Comando INSERT

```
EXEC SQL INSERT INTO DBADB2.TBCADDB2
```

```
( CAD_MATRIC,  
  CAD_NOME,  
  CAD_CURSO,  
  CAD_TURMA,  
  CAD_END,  
  CAD_BAIRRO,  
  CAD_CIDADE,  
  CAD_ESTADO,  
  CAD_TELEF,  
  CAD_EMAIL  
)
```

```
VALUES( :CAD-MATRIC,  
       :CAD-NOME,  
       :CAD-CURSO,  
       :CAD-TURMA,  
       :CAD-END,  
       :CAD-BAIRRO,  
       :CAD-CIDADE,  
       :CAD-ESTADO,  
       :CAD-TELEF  
       :CAD-EMAIL  
)
```

*OS : INDICA UMA VARIÁVEL HOST-Q ESTOU INDO BUSCAR NO COBOL

```
END-EXEC
```

2 – Comando UPDATE:

```
EXEC SQL  
  
UPDATE DBADB2.TBCADDB2  
  
SET  
  
CAD_NOME   =:CAD-NOME,  
CAD_CURSO  =:CAD-CURSO,  
CAD_TURMA  =:CAD-TURMA,  
CAD_END    =:CAD-END,  
CAD_BAIRRO =:CAD-BAIRRO,  
CAD_CIDADE =:CAD-CIDADE,  
CAD_ESTADO =:CAD-ESTADO,  
CAD_TELEF  =:CAD-TELEF,  
CAD_EMAIL  =:CAD-EMAIL  
  
WHERE CAD_MATRIC =:CAD-MATRIC  
  
END-EXEC
```

3 – Comando DELETE:

```
EXEC SQL DELETE  
  
FROM DBADB2.TBCADALUP  
  
WHERE CAD_MATRIC = :CAD-MATRIC  
  
END-EXEC
```

4 – Comando SELECT simples:

EXEC SQL

```
SELECT CAD_MATRIC,  
       CAD_NOME,  
       CAD_CURSO,  
       CAD_TURMA,  
       CAD_END,  
       CAD_BAIRRO,  
       CAD_CIDADE,  
       CAD_ESTADO,  
       CAD_TELEF,  
       CAD_EMAIL  
INTO :CAD-MATRIC,  
     :CAD-NOME,  
     :CAD-CURSO,  
     :CAD-TURMA,  
     :CAD-END,  
     :CAD-BAIRRO,  
     :CAD-CIDADE,  
     :CAD-ESTADO,  
     :CAD-TELEF,  
     :CAD-EMAIL  
FROM DBADB2.TBCADDB2  
WHERE CAD_MATRIC = :CAD-MATRIC  
END-EXEC.
```

5 – Comando SELECT com cursor:

```
EXEC SQL DECLARE CUR1 CURSOR FOR  
SELECT CAD_MATRIC,  
CAD_NOME,  
CAD_CURSO, CAD_TURMA,  
CAD_END,  
CAD_BAIRRO,  
CAD_CIDADE,  
CAD_ESTADO,  
CAD_TELEF,  
CAD_EMAIL  
FROM DBADB2.TBCADALUP  
WHERE CAD_MATRIC > :CAD-MATRIC  
ORDER BY CAD_MATRIC  
END-EXEC.
```

```
EXEC SQL DECLARE CUR2 CURSOR FOR  
SELECT CAD_MATRIC,  
CAD_NOME,  
CAD_CURSO,  
CAD_TURMA,  
CAD_END,  
CAD_BAIRRO,  
CAD_CIDADE,  
CAD_ESTADO,  
CAD_TELEF,  
CAD_EMAIL  
FROM DBADB2.TBCADALUP
```



```
WHERE CAD_MATRIC < :CAD-MATRIC
```

```
ORDER BY CAD_MATRIC DESC
```

```
END-EXEC.
```

```
EXEC SQL
```

```
OPEN CUR1
```

```
END-EXEC.
```

```
EXEC SQL
```

```
CLOSE CUR1
```

```
END-EXEC.
```

```
EXEC SQL FETCH CUR1
```

```
INTO :CAD-MATRIC,
```

```
:CAD-NOME,
```

```
:CAD-CURSO,
```

```
:CAD-TURMA,
```

```
:CAD-END,
```

```
:CAD-BAIRRO,
```

```
:CAD-CIDADE,
```

```
:CAD-ESTADO,
```

```
:CAD-TELEF,
```

```
:CAD-EMAIL
```

```
END-EXEC.
```

6 - Tratamento de campo NULO na tabela DB2.

05 WRK-NU-CPF-ALUNO-N PIC S9(04) COMP VALUE ZEROS.

EXEC SQL

FETCH CSR01-CURSOR

INTO :FUALU.CO-FUTU-T01

,:FUALU.NU-CPF-ALUNO

:WRK-NU-CPF-ALUNO-N

,:FUALU.NO-ALUNO

,:FUALU.ED-ALUNO

,:FUALU.NT-ALUNO

END-EXEC.

IF NU-CPF-ALUNO-N LESS ZEROS

MOVE ZEROS TO U-CPF-ALUNO OF FUALU

END-IF.

7 – JOIN de tabelas:

