

```

SET SERVEROUTPUT ON

DROP TABLE ALUNO CASCADE CONSTRAINTS;

CREATE TABLE ALUNO (
RA NUMBER(4),
NOTA NUMBER(3,1));

INSERT INTO ALUNO VALUES (1001,5);
INSERT INTO ALUNO VALUES (1002,6);
INSERT INTO ALUNO VALUES (1003,8);

--IF-THEN-ELSE
--Executa um conjunto de ações de acordo com uma ou mais condições.
DECLARE
  V_RA ALUNO.RA%TYPE := 1001;
  V_NOTA ALUNO.NOTA%TYPE;
  V_CONCEITO VARCHAR2(12);
BEGIN
  SELECT NOTA INTO V_NOTA FROM ALUNO WHERE RA = V_RA;
  IF V_NOTA <= 5 THEN V_CONCEITO := 'REGULAR';
  ELSIF V_NOTA < 7 THEN V_CONCEITO := 'BOM';
  ELSE V_CONCEITO := 'EXCELENTE';
  END IF;
  DBMS_OUTPUT.PUT_LINE (V_CONCEITO);
END;
/

--IF-THEN-ELSE
--COM &RA_ALUNO
DECLARE
  V_RA ALUNO.RA%TYPE := &RA_ALUNO;
  V_NOTA ALUNO.NOTA%TYPE;
  V_CONCEITO VARCHAR2(12);
BEGIN
  SELECT NOTA INTO V_NOTA FROM ALUNO WHERE RA = V_RA;
  IF V_NOTA <= 5 THEN V_CONCEITO := 'REGULAR';
  ELSIF V_NOTA < 7 THEN V_CONCEITO := 'BOM';
  ELSE V_CONCEITO := 'EXCELENTE';
  END IF;
  DBMS_OUTPUT.PUT_LINE (V_CONCEITO);
END;
/

--CASE
--Retorna determinado resultado de acordo com o valor da variável de comparação.
DECLARE
  V_RA ALUNO.RA%TYPE := 1;
  V_NOTA ALUNO.NOTA%TYPE;
  V_CONCEITO VARCHAR2(12);
BEGIN
  SELECT NOTA INTO V_NOTA FROM ALUNO WHERE RA = V_RA;
  V_CONCEITO :=
  CASE
    WHEN V_NOTA <= 5 THEN 'REGULAR'
    WHEN V_NOTA < 7 THEN 'BOM'
    ELSE 'EXCELENTE'
  END;
  DBMS_OUTPUT.PUT_LINE (V_CONCEITO);
END;
/

```

```

--LOOP
--Repete um bloco de comandos até que a variável contadora atinja o valor final.
--A variável contadora não deve ser declarada na seção DECLARE
--e deixará de existir após a execução do comando END LOOP.
DECLARE
    V_AUXILIAR NUMBER(2) := 0;
BEGIN
    FOR V_CONTADOR IN 1..10
    LOOP
        V_AUXILIAR := V_AUXILIAR +1;
        DBMS_OUTPUT.PUT_LINE (V_AUXILIAR);
    END LOOP;
END;
/

--LOOP
DECLARE
    V_RA_INICIAL ALUNO.RA%TYPE := 1;
    V_RA_FINAL V_RA_INICIAL%TYPE;
    V_AUXILIAR V_RA_INICIAL%TYPE := 0;
BEGIN
    SELECT COUNT(RA)
    INTO V_RA_FINAL
    FROM ALUNO;
    FOR V_CONTADOR IN V_RA_INICIAL..V_RA_FINAL
    LOOP
        V_AUX := V_AUXILIAR +1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE ('Total de alunos: ' || V_AUXILIAR);
END;
/

--WHILE
--Repete um bloco de comandos enquanto a condição no comando WHILE for verdadeira.
DECLARE
    V_AUXILIAR NUMBER(2) := 0;
BEGIN
    WHILE V_AUXILIAR < 10
    LOOP
        V_AUXILIAR := V_AUXILIAR +1;
        DBMS_OUTPUT.PUT_LINE (V_AUXILIAR);
    END LOOP;
END;
/

--WHILE
DECLARE
    V_RA_FINAL ALUNO.RA%TYPE := 1;
    V_AUX V_RA_FINAL%TYPE := 0;
BEGIN
    SELECT COUNT(RA)
    INTO V_RA_FINAL
    FROM ALUNO;
    WHILE V_AUX < V_RA_FINAL
    LOOP
        V_AUX := V_AUX +1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE ('Total de alunos: ' || V_AUX);
END;
/

```

```

--EXIT
--EXEMPLO: LOOP COM EXIT
DECLARE
  V_AUX NUMBER(2) := 0;
BEGIN
  FOR V_CONTADOR IN 1..15
  LOOP
    V_AUX := V_AUX +1;
    DBMS_OUTPUT.PUT_LINE (V_AUX);
    IF V_CONTADOR = 10
    THEN EXIT;
    END IF;
  END LOOP;
END;
/

--TRATAMENTO DE EXCEÇÕES
--EXCEÇÕES PREDEFINIDAS
DECLARE
  V_RA ALUNO.RA%TYPE;
  V_NOTA ALUNO.NOTA%TYPE;
BEGIN
  SELECT RA, NOME INTO V_RA, V_NOTA FROM ALUNO WHERE RA=1001;
  DBMS_OUTPUT.PUT_LINE(V_RA || ' - ' || V_NOME);
EXCEPTION
  WHEN NO_DATA_FOUND
    THEN DBMS_OUTPUT.PUT_LINE ('Não há nenhum aluno com este RA');
  WHEN TOO_MANY_ROWS
    THEN DBMS_OUTPUT.PUT_LINE ('Há mais de um aluno com este RA');
  WHEN OTHERS
    THEN DBMS_OUTPUT.PUT_LINE ('Erro desconhecido');
END;
/

--TRATAMENTO DE EXCEÇÕES
--EXCEÇÕES DEFINIDAS PELO USUÁRIO
DECLARE
  V_RA ALUNO.RA%TYPE := &RA_ALUNO;
  V_NOTA ALUNO.NOTA%TYPE := &NOTA_ALUNO;
  V_CONTA NUMBER(2);
  ALUNO_EXCEPT EXCEPTION;
BEGIN
  SELECT COUNT(RA) INTO V_CONTA FROM ALUNO;
  IF V_CONTA = 5 THEN
    RAISE ALUNO_EXCEPT;
  ELSE INSERT INTO ALUNO VALUES (V_RA,V_NOTA);
  END IF;
EXCEPTION
  WHEN ALUNO_EXCEPT THEN DBMS_OUTPUT.PUT_LINE('Turma cheia!');
END;
/

```