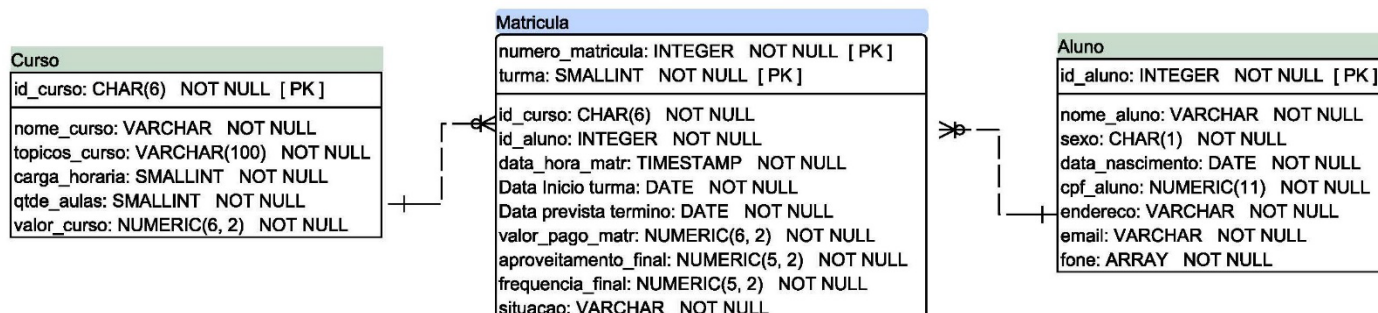
	ECM978 – MATUTINO	
	1º. semestre – 1ª Avaliação	
Disc.: EMC978- BANCOS DE DADOS NÃO CONVENCIONAIS		
Curso: ENGENHARIA DA COMPUTAÇÃO	Série: 5º ANO	

Utilize o Diagrama Lógico-Relacional PARCIAL desenvolvido para uma aplicação que controla as matrículas em cursos de atualização tecnológica.



1- (2,0) Implemente este banco como **Objeto-Relacional no Oracle**, utilizando o **APEX** ou o **Oracle local**, agora considerando que um aluno pode ter mais de um telefone. Adicione os alunos matriculados em determinado curso, por exemplo Python Avançado com as matrículas 100 do aluno José Silva e 101 da aluna Rita Santos etc.

Insira um curso, e duas matrículas-alunos para este curso. Não precisa colocar todos os atributos de matrícula e aluno, só os que interessam para responder às consultas da questão 2. A ideia é que ao consultar um curso suas matrículas venham junto. (*Obs.: quanto mais próxima a solução parecer relacional menor o valor atribuído*)

```

-- definindo um vetor de telefones
DROP TYPE tipofone FORCE ;
CREATE TYPE tipofone AS VARRAY(3) OF INTEGER ;
-- definindo um vetor de topicos de curso
DROP TYPE tipotopico FORCE ;
CREATE TYPE tipotopico AS VARRAY(20) OF VARCHAR2(25) ;
  
```

```

-- tipo para endereço
DROP TYPE tipoendereco FORCE;
CREATE OR REPLACE TYPE tipoendereco AS OBJECT
( logradouro VARCHAR2(50),
  numero SMALLINT,
  complemento CHAR(10),
  bairro VARCHAR2(25),
  cidade VARCHAR2(25),
  UF CHAR(2) ,
  CEP INTEGER ) FINAL;
  
```

```

DROP TYPE tipoaluno FORCE ;
CREATE TYPE tipoaluno AS OBJECT
( id_aluno INTEGER,
  nome_aluno VARCHAR2(40),
  endereco tipoendereco,
  fone tipofone,
  email VARCHAR2(32),
  sexo CHAR(1),
  dt_nascto DATE ) FINAL ;
  
```

```

-- tabela tipada de aluno
DROP TABLE or_aluno CASCADE CONSTRAINTS;
CREATE TABLE or_aluno OF tipoaluno
( PRIMARY KEY ( id_aluno),
nome_aluno NOT NULL,
CHECK ( sexo IN ('M', 'F') ) )
OBJECT IDENTIFIER IS SYSTEM GENERATED ;

-- populando aluno
INSERT INTO or_aluno VALUES ( 1, 'Tiao Macale', tipoendereco('Rua Vergueiro', 7100, null,
'Vila Nair', 'Sao Paulo', 'SP', 09876543),
tipofone (11987, 11654, 11321 ) , 'macale@gmail.com', 'M', '10/10/1990');
INSERT INTO or_aluno VALUES ( 2, 'Abgail Miranda', tipoendereco('Av Nazare', 1100, null,
'Ipiranga', 'Sao Paulo', 'SP', 01234567),
tipofone (118899, 117766, 115544 ) , 'abmiranda@gmail.com', 'F', '03/20/2001');

SELECT * FROM or_aluno ;
SELECT a.nome_aluno, a.endereco.logradouro, a.endereco.numero, f.*
FROM or_aluno a, TABLE(a.fone) f ;

-- tipo para curso
DROP TYPE tipocurso FORCE ;
CREATE OR REPLACE TYPE tipocurso AS OBJECT
( id_curso SMALLINT,
nome_curso VARCHAR2(40),
valor_curso NUMBER(10,2),
carga_horaria SMALLINT,
qtde_vagas INTEGER,
topicos tipotopico ) FINAL ;

-- tabela tipada de curso
DROP TABLE or_curso CASCADE CONSTRAINT;
CREATE TABLE or_curso OF tipocurso
( PRIMARY KEY ( id_curso),
nome_curso NOT NULL )
OBJECT IDENTIFIER IS SYSTEM GENERATED ;

-- populando
INSERT INTO or_curso (id_curso, nome_curso, valor_curso, carga_horaria, qtde_vagas, topicos)
VALUES ( 1, 'SQL Básico', 500, 16, 30, tipotopico ('Criando tabelas', 'Manipulando dados', 'Consultas simples')
);
INSERT INTO or_curso (id_curso, nome_curso, valor_curso, carga_horaria, qtde_vagas, topicos)
VALUES ( 2, 'Python Avançado', 1020, 20, 30, tipotopico ('Manipulando listas', 'Recursividade', 'Funcao Lambda')
);

SELECT c.*, t.*
FROM or_curso c , TABLE ( c.topicos) t ;

-- tipo para matricula
DROP TYPE tipomatrícula FORCE;
CREATE OR REPLACE TYPE tipomatrícula AS OBJECT
( num_matr INTEGER,
turma CHAR(5),
dt_hora_matricula TIMESTAMP,
dt_ini_turma DATE,
valor_pago_matr NUMBER(10,2),

```

```

aproveitamento NUMBER(5,2),
frequencia NUMBER(5,2),
aluno REF tipoaluno ) ;

-- tabela para tipo matricula
DROP TYPE tb_matricula FORCE;
CREATE OR REPLACE TYPE tb_matricula AS TABLE OF tipomatrícula ;

-- criando o novo atributo para as matriculas em curso
ALTER TYPE tipocurso ADD ATTRIBUTE ( matriculas tb_matricula)
CASCADE NOT INCLUDING TABLE DATA ;

-- populando matriculas no curso
UPDATE or_curso
SET matriculas = tb_matricula
( tipomatrícula ( 23000, 'N023', current_timestamp - 22, current_date -15 , 975 , 78, 89,
  ( SELECT REF(a) FROM or_aluno a WHERE a.id_aluno = 1 )) ,
  tipomatrícula ( 23001, 'N023', current_timestamp - 20, current_date -15 , 1050 , 88, 80,
  ( SELECT REF(a) FROM or_aluno a WHERE a.id_aluno = 2 )))
WHERE id_curso = 2 ;
UPDATE or_curso
SET matriculas = tb_matricula
( tipomatrícula ( 23002, 'D011', current_timestamp - 12, current_date + 5 , 650 , 75, 75,
  ( SELECT REF(a) FROM or_aluno a WHERE a.id_aluno = 1 )) ,
  tipomatrícula ( 23003, 'D011', current_timestamp - 10, current_date + 5 , 585 , 75, 75,
  ( SELECT REF(a) FROM or_aluno a WHERE a.id_aluno = 2 )))
WHERE id_curso = 1 ;

-- verificando curso
SELECT c.nome_curso, matr.aluno.nome_aluno, matr.aluno.endereco.logradouro, fn.*
FROM or_curso c, TABLE (c.matriculas) matr, TABLE (matr.aluno.fone) fn ;

```

2- (1,0) Crie um método para o tipo Aluno que retorne a idade como inteiro.

```
CREATE OR REPLACE FUNCTION get_idade ( aluno IN INTEGER)
RETURN INTEGER
AS
v_idade INTEGER := 0 ;
BEGIN
    SELECT ROUND((current_date - a.dt_nascto)/365.25,0) INTO v_idade
    FROM or_aluno a
        WHERE a.id_aluno = aluno ;
RETURN v_idade ;
END ;

-- declarando como método de aluno
ALTER TYPE tipoaluno
ADD MEMBER FUNCTION get_idade ( aluno IN INTEGER)
RETURN INTEGER CASCADE ;

CREATE OR REPLACE TYPE BODY tipoaluno IS
MEMBER FUNCTION get_idade ( aluno IN INTEGER)
RETURN INTEGER
AS
v_idade INTEGER := 0 ;
BEGIN
    SELECT ROUND((current_date - a.dt_nascto)/365.25,0) INTO v_idade
    FROM or_aluno u
        WHERE a.id_aluno = aluno ;
RETURN v_idade ;
END ;
END ;

-- teste
SELECT a.nome_aluno, get_idade(a.id_aluno) AS Idade
FROM or_aluno a ;
```

3 – (1,0) (0,5 pto cada) Responda à seguintes consultas:

- a) Mostre uma lista dos alunos matriculados em cursos de Python no seguinte formato :
- Nome curso – Carga Horária – Numero Matricula – Turma - Nome Aluno – Data Inicio Turma

```
SELECT c.nome_curso, c.carga_horaria, mtr.num_matr, mtr.turma, mtr.aluno.nome_aluno, mtr.dt_ini_turma
FROM or_curso c, TABLE (c.matriculas) mtr
WHERE UPPER(c.nome_curso) LIKE '%PYTHON%';
```

- b) Mostre os alunos com idade entre 25 e 35 anos que tiveram aproveitamento final inferior a 65 para cursos com valor superior a R\$ 1 mil.

```
SELECT mtr.aluno.nome_aluno, get_idade(mtr.aluno.id_aluno) AS Idade
FROM or_curso c, TABLE (c.matriculas) mtr
WHERE mtr.aproveitamento < 65
AND c.valor_curso > 1000
AND get_idade(mtr.aluno.id_aluno) BETWEEN 25 AND 33 ;
```

4- (1,5) Crie o mesmo banco no **MongoDB**. Insira 1 curso com 2 matrículas de alunos.

Solução 1 – 2 coleções : curso e aluno com as matrículas dentro do curso

```
use curso1
show collections
db.aluno.drop()
db.aluno.insert({id: 1, nome: 'Joao da Silva', sexo:'Masculino',
                  Dt_nascimento:ISODate("1990-10-12T04:00:00.000Z"), fones: [1234,5678]})
db.aluno.find()
// inserindo mais de um aluno:
db.aluno.insert([ {id: 2, nome: 'Maria Conceicao', Dt_nascimento:ISODate("1987-10-12T04:00:00.000Z"),
                  cpf: 123456, endereco:'Rua Alfa, 10'},
                  {id:3, nome: 'Rita Andrade', sexo: 'Feminino', Dt_nascimento:ISODate("1999-10-12T04:00:00.000Z"),
                  fones: [123000, 987000] }])

// curso
db.curso.drop()
db.curso.insertMany([ {id: 'PYTAVN', nome:'Python Avancado', topicos : ['Estruturas de Controle', 'Listas'],
                      carga_horaria: 40, qtde_aulas : 10 valor : 1000 },
                      {id: 'SQLAVN', nome:'SQL Avancado', topicos : ['Juncao Externa', 'Consultas aninhadas'],
                      carga_horaria: 20, qtde_aulas : 65 valor : 700 }])
db.curso.find()

// incluindo as matriculas nos cursos
db.curso.updateOne({id:'PYTAVN'},
{$set : {matriculas : [{num_matr: 500, aluno: 1, data_hora: ISODate("2023-02-11T20:00:00.000Z"),
                      turma: 9, aproveitamento_final: 74, freq_final: 79},
                      {num_matr: 501, aluno: 3, data_hora: ISODate("2023-02-12T10:00:00.000Z"),
                      turma: 9, aproveitamento_final: 63, freq_final: 89} ] }})
db.curso.updateOne({id:'SQLAVN'},
{$set : {matriculas : [{num_matr: 622, aluno: 2, data_hora: ISODate("2023-03-21T20:00:00.000Z"),
                      turma: 5, data_ini_turma : ISODate("2023-04-01T19:00:00.000Z"),
                      aproveitamento_final: 84, freq_final: 79},
                      {num_matr: 629, aluno: 3, data_hora: ISODate("2023-03-22T10:00:00.000Z"),
                      turma: 5, data_ini_turma : ISODate("2023-04-01T19:00:00.000Z"),
                      aproveitamento_final: 83, freq_final: 80} ] }})
```

5- (0,5) No **MongoDB** mostre uma lista com os cursos de Python com carga horário superior 20 horas : Nome Curso – Carga Horária – Valor Curso

```
db.curso.find({nome: /python/i , carga_horaria : {$gte: 20}},
              {nome: 1, carga_horaria : 1, valor : 1 } })
```