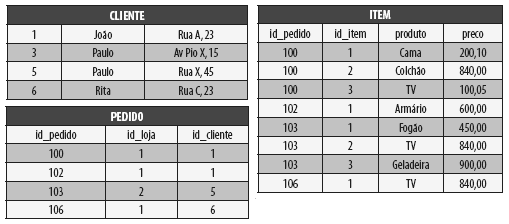
|  |  |  |  |
| --- | --- | --- | --- |
| **Universidade Luterana do Brasil**  **ULBRA – Campus Torres**  **Pró-Reitoria de Graduação**  rosa | | Avaliação Semestral | |
| Curso: STADS | Disciplina: Fundamentos de Bancos de Dados | | Ano/Semestre |
| Turma: quarta/quinta | Professores: Cássio Huggentobler Costa | | 2022-2 |
|  | | |

1. Atente as tabelas abaixo.



Pode-se afirmar que os relacionamentos entre as tabelas Cliente e Pedido e entre as tabelas Pedido e Item são, respectivamente:

1. 1:1 e 1:N
2. 1:N e 1:N
3. 1:N e 1:1
4. **1:N e N:N**
5. N:N e 1:N

Marque a resposta certa e explique por que as outras estão erradas.

1. Incorreta, pois o cliente pode ter mais de 1 pedido conforme a tabela.
2. Incorreta. pois 1 item pode estar em mais de 1 pedido e 1 pedido pode ter vários itens.

c) Pedidos podem ter mais de 1 item, então incorreta.

e) Pedido pode ter 1 e somente 1 cliente.

**Para a resolução dos exercícios subsequentes considere o esquema de banco de dados, nomeado “Acadêmico“, descrito abaixo:**

Professor (idprof, nome, email, endereço, cpf, idcid)

idcid referencia cidade(idcid)

Aluno (idaluno, nome, endereço, telefone, idcid)

idcid referencia cidade(idcid)

Cidade(idcid, nome, UF)

Turma(idturma, idprof, sala, capacidade, idcurso)

idcurso referencia Curso (idcurso)

idprof referencia Professor (idprof)

Disciplina(iddidisc, nome)

Turdisc(idturma, iddisc)

idturma referencia Turma(idturma)

iddisc referencia Disciplina(iddisc)

Matricula(idaluno, idturma)

idaluno referencia Aluno (idaluno)

idturma referencia turma (idturma)

Curso (idcurso, nomecurso, valorcredito)

2. Com base no esquema “Acadêmico“, analise as seguintes afirmações;

I - A tabela Aluno precisa ser criada antes da tabela Disciplina,

II - Antes de criar a tabela Turdisc, as tabelas Turma e Disciplina devem estar criadas.

III - A tabela Curso pode ser a última a ser criada,

É correto somente o que se afirma em:

1. I
2. **II**
3. III
4. I e II
5. II e III

Explique por que as outras estão erradas.

**l Tabela Aluno tem ligação apenas com tabela cidade**

**lll Ela precisa ser criada antes da tabela turma, pois o idcurso tem relação com a tabela Turma.**

3. Marque a sentença que permite listar todas os nomes dos alunos juntamente com o nome da cidade onde moram, ordenado pelo nome do aluno.

1. (\_\_\_) Select aluno.nome, cidade.nome from cidade, aluno
2. (\_\_\_) Select aluno.nome, cidade.nome from aluno,cidade where cidade.idcid=aluno.idaluno order by aluno.nome
3. (\_\_\_) Select aluno.nome, cidade.nome from cidade, aluno where cidade.nome=aluno.nome order by aluno.nome
4. (\_\_\_) Select aluno.nome, cidade.nome from cidade, aluno where cidade.idcid=aluno.idcid
5. **(\_X\_)** Select aluno.nome, cidade.nome from cidade, aluno where cidade.idcid=aluno.idcid order by aluno.nome

* Da resposta que você selecionou correta, refatore-a utilizando joins.

**SELECT a.nome, c.nome**

**FROM Cidades c INNER JOIN Alunos a on a.id\_aluno = c.id\_Cid**

**a.nome;**

4. Faça um SQL que permite atualizar o valor do crédito do curso de nome “ADS” com o acréscimo de 10%.

Mostre o SQL que consulta também, antes de atualizar os dados.

**SELECT nome\_curso, valor\_credito, valor\_credito\*1.10 as novo\_valor**

**FROM Cursos**

**WHERE nome\_curso = "ADS"**

5. Marque a sentença que permite listar os códigos(id) das turmas com o nome da disciplina e a sala.

1. **(\_\_\_) Select idturma, nome, sala from turma, turdisc, disciplina where turma.idturma=turdisc.idturma and disciplina.iddisc=turdisc.iddisc**
2. (\_\_\_) Select idturma, nome, sala from turma
3. **(\_X\_\_\_)**  Select idturma, nome, sala from turma, disciplina where turma.idturma=disciplina.iddisc
4. (\_\_\_) Select idturma, nome, sala from turma, disciplina where disciplina.iddisc=turdisc.iddisc
5. (\_\_\_) Select codturma, nome, sala from turma, disciplina, turdisc

* refatore a resposta escolhida como certa, para utilizar JOINS.

SELECT T.id\_turma, D.nome, T.sala from Turmas T INNER join Disciplinas D on T.id\_turma = D.id\_disc

6. Defina o comando SQL para listar o nome do curso e valor do crédito para o curso que tem o valor mais baixo de crédito(valorcredito).

1. (**\_X\_**\_) Select nomecurso, min(valorcredito) from curso
2. (\_\_\_) Select nomecurso, valorcredito from curso where valorcredito=(select min(valorcredito) from curso)
3. (\_) Select nomecurso, max(valorcredito) from curso
4. (\_\_\_) Select nomecurso, valorcredito from curso where min(curso)
5. (**\_\_\_**)Select nomecurso, min(valorcredito) from aluno where min(valorcredito)

* refatore a resposta escolhida como certa, para utilizar JOINS.

**Tanto a) e b) funcionam, porém nenhuma é possível fazer JOINS, pois é somente 1 tabela.**

7. Explique o conceito Integridade Referencial. De um exemplo SQL DDL para ilustrar melhor a sua explicação, usando a modelagem relacional (“Academico”).

Integridade referencial é um conceito de banco de dados que garante que todos os relacionamentos propostos entre tabelas no modelo de entidade-relacionamento (ER) serão respeitados dando a certeza que os dados de um banco de dados estarão íntegros. Esses relacionamentos são baseados nas definições de uma chave primária e uma chave estrangeira, além de regras pré-definidas para a manipulação dessas chaves. As regras definidas foram: um CASCADE para aluno, isto é, se for removido um registro da tabela de aluno, todas as informações sobre endereço, notas ou outros dados que estiverem relacionados a esse determinado aluno serão removidos ou atualizados.

8. Faça um SQL para adicionar a coluna data\_entrada na tabela matricula.

ALTER TABLE Matriculas ADD data\_entrada DATE

9. Faça um SQL para selecionar os alunos que não moram na cidade onde estudam. Considerando que a cidade onde estudam é Torres. Exibir o código, o nome do aluno e a cidade onde ele mora.

SELECT A.id\_aluno, A.endereco

FROM Alunos A INNER JOIN Cidades C on A.endereco <> "TORRES"

10. Faça um SQL para exibir o nome do aluno, o código do curso, o nome do curso e o nome do professor da turma e a data que ele se matriculou. Apresentar somente para alunos que tenham se matriculado entre semestre 1 de 2022, considere que as matrículas iniciam em janeiro.

SELECT A.nome, C.id\_curso, C.nome\_curso, P.nome

FROM Alunos A INNER join Matriculas M

ON A.id\_aluno = M.id\_aluno

INNER join Turmas T

ON M.id\_turma = T.id\_turma

INNER join Professores P

ON T.id\_prof = P.id\_prof

INNER join Cursos C

ON P.id\_prof = C.id\_curso

WHERE M.data\_matricula >'2022-01-01' end M.data\_matricula < '2022-06-30'

GROUP BY M.data\_matricula;

11. Faça um SQL que liste os 2 professores que mais turmas já atenderam. Importante mostrar o nome do professor.

SELECT P.id\_prof, P.nome as Professor

FROM Professores P, Turmas T

GROUP BY P.id\_prof desc,P.nome

HAVING COUNT(\*) > 1

LIMIT 2

SELECT P.id\_prof, P.nome as Professor

FROM Professores P INNER join Turmas T

ON T.id\_prof = P.id\_prof

GROUP BY P.id\_prof desc,P.nome

HAVING COUNT(\*) > 1

LIMIT 2