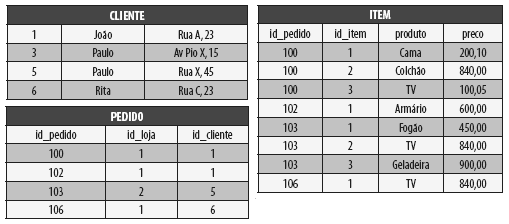
| **Universidade Luterana do Brasil**  **ULBRA – Campus Torres**  **Pró-Reitoria de Graduação**  rosa | | Avaliação Semestral | |
| --- | --- | --- | --- |
| Curso: STADS | Disciplina: Fundamentos de Bancos de Dados | | Ano/Semestre |
| Turma: quarta/quinta | Professores: Cássio Huggentobler Costa |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

Como entregar esta tarefa.

Opções (Escolha uma para a entrega)

1. Fazer uma cópia deste documento e abaixo de cada questão incluir as respostas correspondentes. Se esta opção for a escolhida, coloque a fonte em vermelho e do tipo courier new.
2. Faça um arquivo .sql. Coloque as questões lá em comentário e logo abaixo o sql correspondente. Suba este arquivo para uma pasta no seu drive e entregue o link compartilhado.
3. 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:Nneste caso a cardinalidade da tabela Cliente e pedidos não pode ser 1:1, pois um cliente pode fazer um ou muitos pedidos
2. 1:N e 1:N correta
3. 1:N e 1:1neste caso a cardinalidade da tabela pedido e itens não pode ser 1:1 pois um pedido pode ter um ou muitos itens
4. 1:N e N:N neste caso a cardinalidade da tabela pedido e itens não pode ser N:N pois um pedido pode ter somente um item
5. N:N e 1:Nneste caso a cardinalidade da tabela Cliente e pedidos não pode ser N:N, pois um cliente também pode fazer somente um pedido

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

**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.

A tabela Aluno não precisa ser criada antes da tabela disciplina pois a mesma não é referenciada na tabela disciplina.

A tabela curso não pode ser a última a ser criada pois está sendo referenciada na 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 alunos.nome AS nome\_alunos, cidades.nome AS nome\_cidades

FROM alunos

INNER JOIN cidades ON cidades.id=alunos.idcid

ORDER BY alunos.nome

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

SELECT valor\_credito

FROM cursos

WHERE nome\_curso = 'Análise e Desenvolvimento de Sistemas';

SELECT valor\_credito \* 1.10 as novo\_valor\_credito

FROM cursos

WHERE nome\_curso = 'Análise e Desenvolvimento de Sistemas';

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

1. (\_x\_\_) 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. (\_\_\_) 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 AS id\_turma, d.nome as nome\_disciplina, t.sala as numero\_sala

FROM turmas t

INNER JOIN turdisc td

ON t.id=td.idturma

INNER JOIN disciplinas d

on d.id=td.iddisc

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. (\_\_\_) Select nomecurso, min(valorcredito) from curso
2. (\_X\_\_) 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.

Não é possível refatorar para JOINS pois os dados estão em uma única tabela.

Select nome\_curso, valor\_credito

from cursos

where valor\_credito=(select min(valor\_credito) from cursos);

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 os relacionamentos propostos nas tabelas do modelo ER serão respeitados , dando a garantia de fidelidade dos dados do banco de dados.

Exemplo na pasta no drive em arquivo sql

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

alter table matriculas

add COLUMN 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 as codigo\_aluno, a.nome as nome\_aluno, c.nome as nome\_cidade

from alunos a

INNER join cidades c on a.idcid=c.id

WHERE c.nome <> '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 as nome\_aluno, c.id as codigo\_curso, c.nome\_curso as nome\_curso, p.nome as nome\_professor, m.data\_entrada as data\_matricula

from alunos a

inner join matriculas m on m.idaluno = a.id

inner join turmas t on t.id = m.idturma

inner join cursos c on c.id = t.idcurso

inner join professores p on p.id = t.idprof

WHERE m.data\_entrada BETWEEN '2022-01-01' and '2022-07-10';

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

select p.nome, count(t.id) as quantidade\_turmas\_professor

from professores p

inner join turmas t on p.id = t.idprof

group by p.nome

order by quantidade\_turmas\_professor DESC

limit 2;