|  |
| --- |
|  |

 Olá Turma, Sejam Bem Vindos!

1. Essa avaliação é composta por **10 (QUINZE**) questões de múltipla escolha e abrange os conteúdos dos capítulos 1  e 2.
2. Em cada opção há 5 (cinco) alternativas onde somente 1 (uma) delas é correta.
3. Responda, obrigatoriamente, todas as questões.
   1. Esse questionário irá permanecer aberto até o dia **anterior da nossa aula da próxima semana** ate às **23h:59m**
4. Antes de iniciar informe obrigatoriamente o seu RM (SOMENTE NÚMEROS) e o seu NOME COMPLETO (SEM ABREVIAR) na próxima página.
5. Todas as questões são aplicáveis no servidor de banco de dados Oracle
6. Ao finalizar essa atividade será exibida a seguinte mensagem: **Respostas Enviadas com sucesso. Aguarde a divulgação da notas.**Se a mensagem não for exibida o questionário não foi enviado!

Bom Trabalho!

Prof. Me. Alexandre Barcelos

Parte superior do formulário

Parte superior do formulário

Diagrama

Descrição gerada automaticamente

Parte superior do formulário

p

**Question Title**

\* 3.

Qual dos operadores de conjunto retorna todas as linhas distintas selecionadas pelas duas consultas?

 0

UNION

UNION ALL

INTERSECT

MINUS

EXCEPTParte superior do formulário

p

**Question Title**

\* 4.

Considere o modelo entidade relacionamento a seguir:  
  
Diagrama

Descrição gerada automaticamente  
O modelo de entidade relacionamento apresentado a, representa de forma sucinta uma solução para persistência de dados de uma biblioteca. Considerando que um livro está emprestado quando possuir um registro vinculado a ele na tabela EMPRESTIMO, e essa tupla não possuir um valor na coluna data\_dev, o comando SQL que deve ser utilizado para listar os títulos dos livros disponíveis para empréstimo é:

 0

SELECT titulo FROM livros  
MINUS  
SELECT l.titulo  
FROM   emprestimos e INNER JOIN livros l  
ON     e.cod\_livro = l.cod\_livro  
WHERE  e.data\_dev IS NULL;

SELECT titulo FROM livros  
UNION  
SELECT l.titulo  
FROM   emprestimos e INNER JOIN livros l  
ON     e.cod\_livro = l.cod\_livro  
WHERE  e.data\_dev IS NULL;

SELECT titulo FROM livros  
MINUS  
SELECT l.titulo  
FROM   emprestimos e INNER JOIN livros l  
ON     e.cod\_livro = l.cod\_livro  
WHERE  e.data\_dev IS NOT NULL;

SELECT titulo FROM livros  
UNION  
SELECT l.titulo  
FROM   emprestimos e LEFT JOIN livros l  
ON     e.cod\_livro = l.cod\_livro   
WHERE  e.data\_dev IS NULL;

SELECT titulo FROM livros  
MINUS  
SELECT l.titulo  
FROM   emprestimos e RIGHT JOIN livros l  
ON     e.cod\_livro = l.cod\_livro  
WHERE  e.data\_dev IS NOT NULL ;

Parte superior do formulário

**Question Title**

\* 5. Utilize as tabelas a seguir e suas linhas para responder a questão  
  
TABLE: my\_brick\_collection  
SELECT \* FROM my\_brick\_collection;  
  
COLOUR      SHAPE      WEIGHT  
---------- ---------- ----------  
red        cube               10  
blue       cuboid              8  
green      pyramid            20  
green      pyramid            20  
           cuboid             20  
  
TABLE: your\_brick\_collection  
SELECT \* FROM my\_brick\_collection;  
  
HEIGHT          WIDTH      DEPTH     COLOUR   SHAPE  
---------- ---------- ---------- ---------- -------  
         2          2          2 red        cube  
         2          2          2 blue       cube  
         2          2          8            cuboid  
  
Complete a consulta a seguir, com o operador de conjunto, para retornar uma lista de todas as formas existentes nas duas tabelas.  
  
SELECT shape FROM my\_brick\_collection  
\_\_\_\_\_\_\_\_\_\_\_\_  
SELECT shape FROM your\_brick\_collection  
ORDER  BY shape;  
  
A consulta anterior deve retornar as seguintes linhas:  
  
SHAPE   
cube       
cube       
cube       
cuboid     
cuboid

UNION

UNION ALL

INTERSECT

MINUS

**on Title**

\* 6. Utilize as tabelas a seguir e suas linhas para responder a questão  
  
TABLE: my\_brick\_collection  
SELECT \* FROM my\_brick\_collection;  
  
COLOUR     SHAPE      WEIGHT  
---------- ---------- ----------  
red        cube               10  
blue       cuboid              8  
green      pyramid            20  
green      pyramid            20  
           cuboid             20  
  
TABLE: your\_brick\_collection  
SELECT \* FROM my\_brick\_collection;  
  
HEIGHT     WIDTH      DEPTH      COLOUR     SHAPE  
---------- ---------- ---------- ---------- -------  
         2          2          2 red        cube  
         2          2          2 blue       cube  
         2          2          8            cuboid  
  
Complete a consulta a seguir, com o operador de conjunto, para retornar uma lista de todas as cores existentes nas duas tabelas. Cada cor deve aparecer apenas uma vez:  
  
SELECT colour FROM my\_brick\_collection  
\_\_\_\_\_\_\_\_\_\_\_\_  
SELECT colour FROM your\_brick\_collection  
ORDER BY colour;  
  
A consulta anterior deve retornar as seguintes linhas:  
  
COLOUR    
blue      
green     
red       
<null>  
  
Escolha a alternativa correta:

UNION

UNION ALL

INTERSECT

MINUS

EXCEPT

**stion Title**

\* 7. Utilize as tabelas a seguir e suas linhas para responder a questão  
  
**TABLE: my\_brick\_collection**  
SELECT \* FROM my\_brick\_collection;  
  
COLOUR     SHAPE      WEIGHT  
---------- ---------- ----------  
red        cube               10  
blue       cuboid              8  
green      pyramid            20  
green      pyramid            20  
           cuboid             20  
  
**TABLE: your\_brick\_collection**  
SELECT \* FROM my\_brick\_collection;  
  
HEIGHT     WIDTH       DEPTH     COLOUR     SHAPE  
---------- ---------- ---------- ---------- -------  
         2          2          2 red        cube  
         2          2          2 blue       cube  
         2          2          8            cuboid  
  
Complete a consulta a seguir, com o operador de conjunto, para retornar uma lista de todas as formas somente da minha coleção.  
  
SELECT shape FROM my\_brick\_collection  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
SELECT shape FROM your\_brick\_collection;  
  
A consulta anterior deve retornar a seguinte linha:  
  
SHAPE      
pyramid   
  
Escolha a alternativa correta:  0

UNION

UNION ALL

INTERSECT

MINUS

EXCEPT

**tion Title**

\* 8. Utilize as tabelas a seguir e suas linhas para responder a questão  
  
TABLE: my\_brick\_collection  
SELECT \* FROM my\_brick\_collection;  
  
COLOUR      SHAPE     WEIGHT  
---------- ---------- ----------  
red        cube               10  
blue       cuboid              8  
green      pyramid            20  
green      pyramid            20  
           cuboid             20  
  
TABLE: your\_brick\_collection  
SELECT \* FROM my\_brick\_collection;  
  
HEIGHT     WIDTH      DEPTH      COLOUR     SHAPE  
---------- ---------- ---------- ---------- -------  
         2          2          2 red        cube  
         2          2          2 blue       cube  
         2          2          8            cuboid  
  
Complete a consulta a seguir, com o operador de conjunto, para retornar uma lista de todas as cores que estão nas duas tabelas:  
  
SELECT colour FROM my\_brick\_collection  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
SELECT colour FROM your\_brick\_collection  
ORDER BY colour;  
  
A consulta anterior deve retornar a seguinte linha:  
  
COLOUR    
blue      
red       
<null>    
  
Escolha a alternativa correta:

UNION

UNION ALL

INTERSECT

MINUS

\* 9. Os operadores de conjunto combinam os resultados de duas ou mais consultas componentes em um resultado. As consultas que contêm operadores de conjunto são denominadas consultas compostas.  
  
Escolha a opção que apresenta como o SGBDR Oracle avalia uma instrução SQL SELECT que utiliza operadores de conjunto.  0

Os operadores são avaliados na seguinte ordem: UNION, UNION ALL, MINUS e INTERSECT.

Os operadores são avaliados na seguinte ordem: MINUS, INTERSECT, UNION e UNION ALL.

O operador MINUS tem precedência sobre todos os operadores.

O operador INTERSECT tem precedência sobre todos os operadores.

Todos os operadores de conjunto têm a mesma precedência.

NOVA PERGUNTA

10. Escolha a instrução SQL que mostra todos os departamentos cujo salário mínimo é maior que o do departamento 50.  0

SELECT   department\_id, MIN(salary)  
FROM     employees  
GROUP BY department\_id  
HAVING   MIN(salary) >  
                       (SELECT MIN(salary)  
                        FROM   employees  
                        WHERE  department\_id = 50);

SELECT   department\_id, MIN(salary)  
FROM     employees  
GROUP BY department\_id  
HAVING   MIN(salary) IN  
                       (SELECT MIN(salary)  
                        FROM   employees  
                        WHERE  department\_id = 50);

SELECT   department\_id, MIN(salary)  
FROM     employees  
GROUP BY department\_id  
HAVING   MIN(salary) > ANY  
                       (SELECT MIN(salary)  
                        FROM   employees  
                        WHERE  department\_id = 50);

SELECT   department\_id, MIN(salary)  
FROM     employees  
GROUP BY department\_id  
HAVING   MIN(salary) > ALL  
                       (SELECT MIN(salary)  
                        FROM   employees  
                        WHERE  department\_id = 50);

SELECT   department\_id, MIN(salary)  
FROM     employees  
GROUP BY department\_id  
HAVING   MIN(salary) > IN  
                       (SELECT MIN(salary)  
                        FROM   employees  
                        WHERE  department\_id = 50);

**tle**

\* 11. Assinale a alternativa que cria uma consulta que exiba o nome das regiões, sem repetições (REGION\_NAME) que tem um ou mais países.  0

select region\_name  
from regions  
intersect  
select region\_name  
from regions r join countries c  
on (r.region\_id = c.region\_id);

select region\_name  
from regions  
minus  
select region\_name  
from regions r join countries c  
on (r.region\_id = c.region\_id);

select region\_name  
from regions  
union  
select region\_name  
from regions r join countries c  
on (r.region\_id = c.region\_id);

select region\_name  
from regions  
union all  
select region\_name  
from regions r join countries c  
on (r.region\_id = c.region\_id);

select region\_name  
from regions  
except  
select region\_name  
from regions r join countries c  
on (r.region\_id = c.region\_id);

p

\* 12. Assinale a alternativa que exibe o nome dos departamentos que tem o salário médio acima da média.  0

select department\_name,avg(salary)  
from   employees e join departments d  
on     d.department\_id=e.department\_id  
group  by department\_name  
having avg(salary) > (select avg(salary)  
                      from employees);

select department\_name,avg(salary)  
from   employees e join departments d  
on     d.department\_id=e.department\_id  
having avg(salary) > (select avg(salary)  
                      from employees)  
group  by department\_name;

select department\_name,avg(salary)  
from   employees e join departments d  
on     d.department\_id=e.department\_id  
where   salary  > (select avg(salary)  
                      from employees)  
group  by department\_name;

select department\_name,avg(salary)  
from   employees e join departments d  
on     d.department\_id=e.department\_id  
group  by department\_name  
where   salary  > (select avg(salary)  
                      from employees);

select department\_name,avg(salary)  
from   employees e join departments d  
on     d.department\_id=e.department\_id  
where salary  >any (select salary  
                      from employees)  
group  by department\_name;

ANTERIOR CONCLUÍDO

Parte inferior do formulário