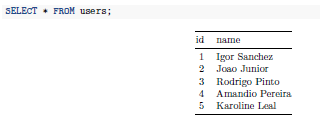
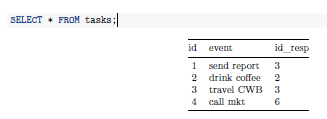
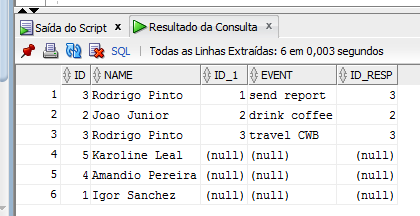
**1. Dadas as tabelas abaixo:**

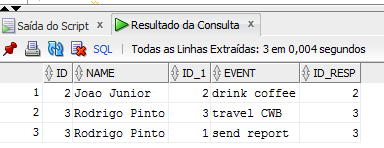




Qual o resultado da query abaixo?

**SELECT** \* **FROM** users **LEFT JOIN** tasks **ON** users.id = tasks.id\_resp;

Irá retornar todos os dados da tabela users que possuem correspondentes na tabela tasks mais os users q não possuem.



**Caso deseje trazer apenas os usuários que possuem tasks**

**SELECT** \* **FROM** HR.USERS

**INNER JOIN** HR.TASKS **ON** HR.USERS.id = HR.TASKS.id\_resp;

**2. Espeficique cada tipo de JOIN:**

• **Left Join –** o left join retorna todos os dados da tabela 1 mesmo que não haja um “match”, ou seja, que por exemplo o id da tabela da esquerda não seja encontrado na tabela da direita.

• **Right Join -** o right join retorna todos os dados da tabela 2 mesmo que não haja um “match”, ou seja, que por exemplo o id da tabela da direita não seja encontrado na tabela da esquerda.

• **Inner Join –** o inner join retorna apenas as linhas de ambas as tabelas na qual houve um “match”, ou seja, o id tabela 1 é igual ao id da tabela 2, caso contrário a linha não é retornada.

• **Full Join -** o full join retorna absolutamente tudo de ambas as tabelas, mesmo que não haja “match” entre as colunas fornecidas para o join.

**3. O que é uma chave primária de uma tabela?**

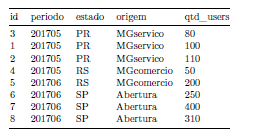
A chave primária identifica cada registro de maneira única em um banco de dados, estes valor devem ser únicos e não nulos.

**4. Quais dessas funções são destinadas a agregação de dados, GROUP BY?**

LEN(), **RIGHT**(), SUM(), **REPLACE**(), COUNT(), CONCAT(), ABS()

Apenas as funções SUM() e COUNT()

**5. Dado a tabela:**



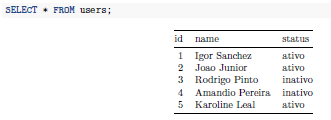
a. Escreva a clausula WHERE que retorne as quantidades do **período 201705** para o **estado do PR** quando as quantidades forem **superiores a 80**.

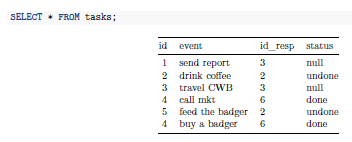
WHERE periodo = 201705   
AND estado = ‘PR’  
AND qtd\_users > 80

b. Quais id linhas serão retornadas?

Serão retornados apenas as linhas com os ids(1,2)

**6. Dadas as tabelas abaixo:**



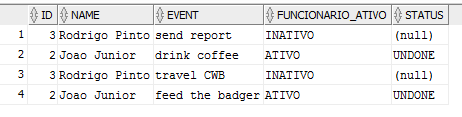


a. Faça uma query contendo o resultado das duas tabelas juntas, renomenando o campo status da tabela users para funcionario\_ativo.

**SELECT** HR.USERS.ID, HR.USERS.NAME, HR.TASKS.EVENT, HR.USERS.STATUS **AS** FUNCIONARIO\_ATIVO, HR.TASKS.STATUS

**FROM** HR.USERS

**INNER** **JOIN** HR.TASKS **ON** HR.USERS.id = HR.TASKS.id\_resp;



b. Faça outra query que traga os eventos com o nome do responsável. O resultado não deve trazer os campos de status de ambas tabelas, porém deve trazer um novo campo de status\_do\_evento que deve construindo da seguinte forma:

• se o status do funcionário for ativo e o status do evento for done, marcar como sucesso

• se o status do funcionário for ativo e o status do evento for done, marcar como falha

• se o status do funcionário for inativo e o status do evento for nulo, marcar como reatribuir

**SELECT** HR.USERS.ID,

HR.USERS.NAME,

HR.TASKS.EVENT,

**CASE**

**WHEN** HR.USERS.STATUS = 'ATIVO' **AND** HR.TASKS.STATUS = 'DONE' **THEN** 'SUCESSO'

**WHEN** HR.USERS.STATUS = 'ATIVO' **AND** HR.TASKS.STATUS = 'UNDONE' **THEN** 'FALHA'

**ELSE** 'REATRIBUIR'

**END** **AS** STATUS\_DO\_EVENTO

**FROM** HR.USERS

**INNER** **JOIN** HR.TASKS **ON** HR.USERS.id = HR.TASKS.id\_resp;

