Obs.: relatório de movimentações Financeiras (A banca só deveria ser calculada se Resultado <> vazio

**Botão Recalcular Plano**

1. Primeira verificação
   1. Verifica (Select)¹ para pegar todas as **Orientações**
      1. Qualificação com Mês/Ano = à data de realização da Quali e “resultado” = “Aprovado ou Reprovado” (isso só na nova versão) ;
      2. Defesa com Mês/Ano = à Data Aprovação Orientador (data\_entrega\_trabalho) ;
   2. Verifica (Select) ³ para pegar todas as participações em **Bancas** e “resultado” = “Aprovado ou Reprovado” (isso só na nova versão) do professor no mês / ano.
   3. Verifica (Select) ² para pegar todas as **Horas/Aulas** com presença do professor no mês / ano.
2. Zerando valores
   1. Faz uma atualização (Update) na tabela **professor\_plano** e coloca (seta) 0 para todos os registros daquele professor naquele mês / ano.
3. Atualizando valores
   1. Verifica se tem algum registro na tabela **professor\_plano** daquele professor naquele mês / ano com motivo = “**Horas Aula**”
      1. Se não então faz uma inserção (Insert)na tabela **professor\_plano** com o valor total e motivo “**Horas Aula**”
      2. Se sim então faz uma atualização (Update)na tabela **professor\_plano** com o valor = valor + Valor Total e motivo “**Horas Aula**”
   2. Verifica se tem algum registro na tabela **professor\_plano** daquele professor naquele mês / ano com motivo = “**Bancas**”
      1. Se não então faz uma inserção (Insert)na tabela **professor\_plano** com o valor total e motivo “**Bancas**”
      2. Se sim então faz uma atualização (Update)na tabela **professor\_plano** com o valor = valor + Valor Total e motivo “**Bancas**”
   3. Verifica se tem algum registro na tabela **professor\_plano** daquele professor naquele mês / ano com motivo = “**Orientação**”
      1. Se não então faz uma inserção (Insert)na tabela **professor\_plano** com o valor total e motivo “**Orientação**”
      2. Se sim então faz uma atualização (Update)na tabela **professor\_plano** com o valor = valor + Valor Total e motivo “**Orientação**”

Anexo

1ª) Select que seleciona os professores que realizaram “Orientações” e “Co-orientações” em bancas de Qualificação e de Defesa.

SELECT \* FROM (

SELECT bp.id\_professor, b.tipo\_banca, tr.id\_curso, b.horario, p.nome as professor, a.nome as aluno,

fr.id\_forma\_recebimento as recebe\_como, p.id\_titulo, t.nome as titulo\_academico, t.reduzido as titulo\_reduzido,

fr.nome as forma\_recebimento, b.resultado, valor\_hora=(CASE WHEN b.tipo\_banca='defesa' THEN cvo.valor\_defesa ELSE cvo.valor\_qualificacao END),

b.data\_entrega\_trabalho, data\_calculo='01/02/2020', b.id\_banca

FROM banca\_professores bp

INNER JOIN banca b ON bp.id\_banca=b.id\_banca

INNER JOIN matricula\_turma mt ON b.id\_matricula\_turma = mt.id\_matricula\_turma

INNER JOIN turmas tr ON mt.id\_turma=tr.id\_turma

INNER JOIN alunos a ON mt.id\_aluno=a.idaluno

INNER JOIN professores p ON bp.id\_professor=p.id\_professor

LEFT JOIN professores\_forma\_recebimento pfr ON p.id\_professor=pfr.id\_professor

LEFT JOIN forma\_recebimento as fr ON pfr.orientacao=fr.id\_forma\_recebimento

LEFT JOIN titulacao as t ON p.id\_titulo=t.id\_titulacao

LEFT JOIN curso\_valor\_orientacao cvo ON cvo.id\_curso=tr.id\_curso AND fr.id\_forma\_recebimento=cvo.id\_forma\_recebimento

WHERE (bp.tipo\_professor='orientador' OR bp.tipo\_professor='co-orientador') AND isNull(pfr.orientacao, 1)>1

AND bp.id\_professor=1041) derivedTbl

WHERE (CASE tipo\_banca WHEN 'Defesa'

THEN CASE WHEN (YEAR(data\_entrega\_trabalho)=YEAR(data\_calculo) AND MONTH(data\_entrega\_trabalho)=MONTH(data\_calculo)) THEN 1 ELSE 0 END

ELSE CASE WHEN (YEAR(horario)=YEAR(data\_calculo) AND MONTH(horario)=MONTH(data\_calculo)

AND (resultado = 'Aprovado' or resultado = 'Reprovado')

) THEN 1 ELSE 0 END END)=1

ORDER BY professor

1b) Select que seleciona os professores que realizaram “Orientações” e “Co-orientações” em bancas de Qualificação e de Defesa. (Melhor compreensão mas de ruim performance)

SELECT ti.reduzido, p.id\_professor, p.nome, fr.nome forma\_pagto, f.nome empresa, pf.orientacao , a.idaluno, a.nome, b.tipo\_banca, b.horario,

valor\_hora=(CASE WHEN b.tipo\_banca='defesa' THEN cvo.valor\_defesa ELSE cvo.valor\_qualificacao END)

FROM professores p

inner join professores\_forma\_recebimento pf on pf.id\_professor = p.id\_professor

inner join banca\_professores bp on bp.id\_professor = p.id\_professor and (bp.tipo\_professor='orientador' OR bp.tipo\_professor='co-orientador')

inner join banca b on b.id\_banca = bp.id\_banca

inner join matricula\_turma mt on b.id\_matricula\_turma = mt.id\_matricula\_turma

inner join turmas t on t.id\_turma = mt.id\_turma

inner join cursos c on c.id\_curso = t.id\_curso

inner join curso\_valor\_orientacao cvo on cvo.id\_curso = c.id\_curso and cvo.id\_forma\_recebimento = pf.orientacao

--aqui é que está a lentidão

inner join alunos a on a.idaluno = mt.id\_aluno

inner join titulacao ti on p.id\_titulo = ti.id\_titulacao

inner join forma\_recebimento fr on fr.id\_forma\_recebimento = pf.orientacao

inner join fornecedores f on f.id\_fornecedor = pf.id\_fornecedor

WHERE (CASE b.tipo\_banca

WHEN 'Defesa'

THEN CASE WHEN (YEAR(b.data\_entrega\_trabalho)=2020 AND MONTH(b.data\_entrega\_trabalho)=02) THEN 1 ELSE 0 END

ELSE CASE WHEN (YEAR(b.horario)=2020 AND MONTH(b.horario)=02

--AND (resultado = 'Aprovado' or resultado = 'Reprovado')

) THEN 1 ELSE 0 END END)=1

--and p.id\_professor in (1048,1041)

order by p.nome, b.horario, a.nome

1c) (Verifica as orientações e co-orientações dadas por professor já totalizadas)

SELECT p.id\_professor, p.nome,

isnull(sum((CASE WHEN b.tipo\_banca='defesa' THEN cvo.valor\_defesa ELSE cvo.valor\_qualificacao END)), 0.0) as valor

FROM professores p

inner join professores\_forma\_recebimento pf on pf.id\_professor = p.id\_professor

inner join banca\_professores bp on bp.id\_professor = p.id\_professor and (bp.tipo\_professor='orientador' OR bp.tipo\_professor='co-orientador')

inner join banca b on b.id\_banca = bp.id\_banca

inner join matricula\_turma mt on b.id\_matricula\_turma = mt.id\_matricula\_turma

inner join turmas t on t.id\_turma = mt.id\_turma

inner join cursos c on c.id\_curso = t.id\_curso

inner join curso\_valor\_orientacao cvo on cvo.id\_curso = c.id\_curso and cvo.id\_forma\_recebimento = pf.orientacao

WHERE (CASE b.tipo\_banca

WHEN 'Defesa'

THEN CASE WHEN (YEAR(b.data\_entrega\_trabalho)=2020 AND MONTH(b.data\_entrega\_trabalho)=02) THEN 1 ELSE 0 END

ELSE CASE WHEN (YEAR(b.horario)=2020 AND MONTH(b.horario)=02

--AND (resultado = 'Aprovado' or resultado = 'Reprovado')

) THEN 1 ELSE 0 END END)=1

--and p.id\_professor in (1048,1041)

--AND o.id\_oferecimento=2112 AND cd.id\_curso=2

GROUP BY p.id\_professor, p.nome, cvo.valor\_defesa, cvo.valor\_qualificacao

2a (Verifica as horas aula dadas por professor detalhada por datas e curso)

SELECT cd.id\_curso, d.id\_disciplina, dap.id\_professor, da.data\_aula, isnull(sum(dap.hora\_aula), 0.0) as hora\_aula

FROM datas\_aulas\_professor dap

INNER JOIN datas\_aulas da ON dap.id\_aula=da.id\_aula

INNER JOIN oferecimentos o ON da.id\_oferecimento=o.id\_oferecimento

INNER JOIN disciplinas d ON d.id\_disciplina=o.id\_disciplina

INNER JOIN cursos\_disciplinas cd ON o.id\_disciplina=cd.id\_disciplina

INNER JOIN presenca\_professor pp ON dap.id\_aula=pp.id\_aula AND dap.id\_professor=pp.id\_professor

WHERE (YEAR(da.data\_aula) = '2020') AND (MONTH(da.data\_aula) = '2')

AND (pp.presente = '1')

AND dap.id\_professor=1041

--AND o.id\_oferecimento=2112 AND cd.id\_curso=2

GROUP BY cd.id\_curso, d.id\_disciplina, dap.id\_professor, da.data\_aula ORDER BY da.data\_aula

2b (Verifica as horas aula dadas por professor já totalizadas por curso-hora\_aula)

SELECT cd.id\_curso, dap.id\_professor, cvha.valor, isnull(sum(dap.hora\_aula), 0.0) as hora\_aula, cvha.valor \* isnull(sum(dap.hora\_aula), 0.0) valor\_total

FROM datas\_aulas\_professor dap

INNER JOIN datas\_aulas da ON dap.id\_aula=da.id\_aula

INNER JOIN oferecimentos o ON da.id\_oferecimento=o.id\_oferecimento

INNER JOIN cursos\_disciplinas cd ON o.id\_disciplina=cd.id\_disciplina

INNER JOIN presenca\_professor pp ON dap.id\_aula=pp.id\_aula AND dap.id\_professor=pp.id\_professor

inner join professores p on p.id\_professor = dap.id\_professor

inner join professores\_forma\_recebimento pf on pf.id\_professor = dap.id\_professor

inner join forma\_recebimento fh on fh.id\_forma\_recebimento = pf.banca

inner join curso\_valor\_hora\_aula cvha on cvha.id\_forma\_recebimento = pf.horas\_aula and cvha.id\_curso = cd.id\_curso and cvha.id\_titulacao = p.id\_titulo

WHERE (YEAR(da.data\_aula) = '2020') AND (MONTH(da.data\_aula) = '2')

AND (pp.presente = '1')

AND dap.id\_professor=1041

--AND o.id\_oferecimento=2112 AND cd.id\_curso=2

GROUP BY p.nome, cd.id\_curso, dap.id\_professor, cvha.valor order by p.nome

3ª) (Verifica as participações em como membro em Bancas professor detalhada)

SELECT bp.id\_professor, b.tipo\_banca, tr.id\_curso, b.horario, p.nome as professor, a.nome as aluno,

fr.id\_forma\_recebimento as recebe\_como, p.id\_titulo, t.nome as titulo\_academico, t.reduzido as titulo\_reduzido,

fr.nome as forma\_recebimento, valor\_hora=(CASE isNull(fr.id\_forma\_recebimento, 1) WHEN 8 THEN cvb.valor\_sao\_paulo WHEN 9 THEN cvb.valor\_fora\_sao\_paulo ELSE 0.0 END)

FROM banca\_professores bp

INNER JOIN banca b ON bp.id\_banca=b.id\_banca

INNER JOIN matricula\_turma mt ON b.id\_matricula\_turma = mt.id\_matricula\_turma

INNER JOIN turmas tr ON mt.id\_turma=tr.id\_turma

INNER JOIN alunos a ON mt.id\_aluno=a.idaluno

INNER JOIN professores p ON bp.id\_professor=p.id\_professor

LEFT JOIN professores\_forma\_recebimento pfr ON p.id\_professor=pfr.id\_professor

LEFT JOIN forma\_recebimento as fr ON pfr.banca=fr.id\_forma\_recebimento

LEFT JOIN titulacao as t ON p.id\_titulo=t.id\_titulacao

LEFT JOIN curso\_valor\_banca cvb ON tr.id\_curso=cvb.id\_curso

WHERE bp.tipo\_professor='Membro' AND isNull(pfr.banca, 1) > 1

AND (resultado = 'Aprovado' or resultado = 'Reprovado')

AND bp.id\_professor=1041

AND YEAR(b.horario)=2020 AND MONTH(b.horario)=02

ORDER BY professor

3b) (Verifica as participações em bancas professor já totalizadas)

SELECT p.id\_professor, p.nome,

isnull(sum((CASE isNull(pf.banca, 1) WHEN 8 THEN cvb.valor\_sao\_paulo WHEN 9 THEN cvb.valor\_fora\_sao\_paulo ELSE 0.0 END)), 0.0) as valor

FROM professores p

inner join professores\_forma\_recebimento pf on pf.id\_professor = p.id\_professor AND isNull(pf.banca, 1) > 1 -- só aparece com valores

inner join banca\_professores bp on bp.id\_professor = p.id\_professor and (bp.tipo\_professor='membro')

inner join banca b on b.id\_banca = bp.id\_banca

inner join matricula\_turma mt on b.id\_matricula\_turma = mt.id\_matricula\_turma

inner join turmas t on t.id\_turma = mt.id\_turma

inner join cursos c on c.id\_curso = t.id\_curso

inner join curso\_valor\_banca cvb on cvb.id\_curso = c.id\_curso

WHERE YEAR(b.horario)=2020 AND MONTH(b.horario)=02

--AND (resultado = 'Aprovado' or resultado = 'Reprovado')

--and p.id\_professor in (1048,1041)

GROUP BY p.id\_professor, p.nome

order by p.nome

4

UPDATE professor\_plano SET valor=0 WHERE id\_professor=1041 AND YEAR(mes)=2020 AND MONTH(mes)=4;

**Botão Solicitar Pagamento**

1. É feita uma consulta na tabela professor\_solicitacao\_plano com professor\_plano e verifica se tem valor na professor\_plano que não tem valor ou o valor está menor que na professor\_solicitacao\_plano.
2. Então o usuário escolhe os valores que se pretende “solicitar o pagamento” e então esse valores são somados e incluídos um único registro na “professor\_solicitacao\_pagamento” e de pois um registro para cada valor na tabela “professor\_solicitacao\_plano” com a chave acima e a chave da “professor\_plano” . Exemplo 3ª).

1ª) Consulta original

SELECT t.\* FROM

(SELECT p.nome as professor, pp.\*,

valor\_solicitado=ISNULL((SELECT SUM(psp.valor\_solicitado)

FROM professor\_solicitacao\_plano psp

WHERE pp.id\_plano=psp.id\_plano) , 0)

FROM professor\_plano pp

INNER JOIN professores p ON pp.id\_professor=p.id\_professor) t

WHERE (t.valor-t.valor\_solicitado<>0)

AND (t.id\_professor = '1041')

--AND (t.motivo <> 'Banca')

ORDER BY t.id\_professor, t.mes, t.motivo

2ª) Consulta otimizada

SELECT t.\*, ISNULL(t.valor-t.valor\_solicitado, 0) valor\_pagar FROM

(SELECT p.nome as professor, pp.\*,

valor\_solicitado=ISNULL((SELECT SUM(psp.valor\_solicitado)

FROM professor\_solicitacao\_plano psp

WHERE pp.id\_plano=psp.id\_plano) , 0)

FROM professor\_plano pp

INNER JOIN professores p ON pp.id\_professor=p.id\_professor) t

WHERE (t.valor-t.valor\_solicitado<>0)

AND (t.id\_professor = '1041')

AND (t.motivo <> 'Banca')

ORDER BY t.id\_professor, t.mes, t.motivo

INSERT INTO professor\_solicitacao\_pagamento(data\_solicitacao,valor,status,data\_alteracao,usuario)

VALUES ('20200526 11:59:01','6540.00','Solicitado','20200526 11:59:01','admin');SELECT CAST(scope\_identity() AS numeric);

INSERT INTO professor\_solicitacao\_plano (id\_solicitacao, id\_plano, valor\_solicitado)

VALUES (3558, 7227, '960.0000');

INSERT INTO professor\_solicitacao\_plano (id\_solicitacao, id\_plano, valor\_solicitado)

VALUES (3558, 7258, '2700.0000');

INSERT INTO professor\_solicitacao\_plano (id\_solicitacao, id\_plano, valor\_solicitado)

VALUES (3558, 7257, '2880.0000');

**Botão Recebimento Nota Fiscal**

1. Verifica-se nas tabelas plano, solicitação e planoxsolicitação se há registros solicitados com status “Solicitado”

Consulta Original:

SELECT pp.id\_professor, pp.mes, pp.valor, pp.motivo, pp.data\_cadastro, pp.data\_alteracao, pp.usuario, p.nome AS professor, 0 AS solicitacoes,

pspg.status, pspg.data\_solicitacao, pspg.valor valor\_pspg,

pp.id\_plano, psp.valor\_solicitado, psp.id\_solicitacao FROM professor\_plano pp INNER JOIN professores p ON pp.id\_professor=p.id\_professor

INNER JOIN professor\_solicitacao\_plano AS psp ON pp.id\_plano=psp.id\_plano

INNER JOIN professor\_solicitacao\_pagamento as pspg ON psp.id\_solicitacao=pspg.id\_solicitacao

WHERE pp.id\_professor = '1020'

and pspg.status='Solicitado'

ORDER BY pp.id\_professor, pspg.data\_solicitacao, pp.mes

Consulta otimizada:

SELECT pp.id\_professor, pspg.data\_solicitacao, pp.mes, pp.motivo, pp.valor, psp.valor\_solicitado, pspg.valor valor\_total\_solicitado,

pp.data\_cadastro, pp.data\_alteracao, pp.usuario, p.nome AS professor, pp.id\_plano, psp.id\_solicitacao, pspg.data\_pagamento, pspg.data\_recebimento, pspg.nota\_fiscal

FROM professor\_plano pp

INNER JOIN professores p ON pp.id\_professor=p.id\_professor

INNER JOIN professor\_solicitacao\_plano AS psp ON pp.id\_plano=psp.id\_plano

INNER JOIN professor\_solicitacao\_pagamento as pspg ON psp.id\_solicitacao=pspg.id\_solicitacao

WHERE pp.id\_professor = '1020'

and pspg.status='Solicitado'

ORDER BY pp.id\_professor, pspg.data\_solicitacao, pp.mes

1. Atualiza a tabela “professor\_solicitacao\_pagamento”

UPDATE professor\_solicitacao\_pagamento SET status='Pago', nota\_fiscal='1234', data\_recebimento='20200601', data\_pagamento='20200602', usuario='admin'

WHERE id\_solicitacao=3537

UPDATE professor\_solicitacao\_pagamento SET status='Pago', nota\_fiscal='1234', data\_recebimento='20200601', data\_pagamento='20200602', usuario='admin'

WHERE id\_solicitacao=3620