

Banco de dados - MaxPromotor WEB

[Introdução](#)
[Diagrama geral do banco de dados](#)
[Principais Tabelas](#)
[Principais consultas de relatórios existentes](#)
[Select de itens do relatório - Visitados x Não Visitado.](#)
[Select principal do relatório - Visitados x Não Visitado.](#)
[Select relatório - Apuração.](#)
[Select relatório - Objetivo de Metas.](#)
[Select relatório - Produtividade e ociosidade.](#)
[Select relatório - Visitas e Justificativas.](#)

Produto: MaxPromotor



Artigos Relacionados

- [Banco de dados - MaxPromotor WEB](#)

Introdução

Este guia é um recurso conciso e objetivo destinado a facilitar a compreensão e o uso eficaz do nosso banco de dados composto por 279 tabelas, voltado para equipes de implantação. O foco está em habilitar tanto a nossa equipe quanto os clientes a extrair informações valiosas desses dados para a construção de soluções de Business Intelligence (B.I.).

Você encontrará aqui:

- **Diagrama Simplificado do Banco de Dados:** Uma visão geral das tabelas, com ênfase nas mais relevantes para a criação de B.I.
- **Descrição das Tabelas Chave:** Detalhes sobre as funções e inter-relações das tabelas principais.
- **Consultas Padrão para Relatórios:** Exemplos de consultas SQL frequentemente usadas para gerar relatórios.

Este documento tem como objetivo oferecer uma compreensão clara e aplicável do nosso banco de dados, capacitando a equipe a apoiar os clientes na análise de dados e na geração de insights de negócios.

Diagrama geral do banco de dados



LINK DA IMAGEM

https://drive.google.com/file/d/1A3qtVVEEnvCkd_RT3mBjEmSCqeacEs1FX/view?usp=drive_link

Como podemos ver no diagrama do banco de dados, existem muitas tabelas relacionadas, o que torna a complexidade de uma consulta na medida em que se relacionam as informações ainda maior, para tentar simplificar iremos abordar apenas as principais tabelas e citar alguns exemplos de queries utilizadas para emissão dos principais relatórios do produtos, que não devem ser compartilhadas com o cliente, onde visam apenas ilustrar e compreender melhor esses relacionamentos a fim de extrair as principais informações contidas no MaxPromotor.

Principais Tabelas

Tabela	Principal função
prow_visita	Tem como função registrar as visitas realizadas pelos usuários do App do MaxPromotor, sendo uma tabela essencial pelo fato de outras tabelas dependerem da visita para registro das informações.
prow_usuario	Tem como função armazenar as informações cadastrais referente ao usuário do sistema.
prow_ponto_venda	Tem como função armazenar as informações cadastrais referente aos pontos de venda disponíveis para atendimento no sistema.
prow_pedido	Tem como função armazenar as informações dos pedidos que vieram de origem do ERP, através da integração de Pedidos.
prow_pedido_produto	Tem como função armazenar as informações dos itens avaliados do tipo Produto pertencentes a um Pedido.
prow_agenda	Tem como função armazenar as informações da agenda de atendimento do usuário do App do MaxPromotor.
prow_item_agenda	Tem como função armazenar as informações detalhadas vinculadas à agenda de atendimento do usuário do App do MaxPromotor.
prow_pergunta	Tem como função armazenar as informações da Pergunta que podem ser utilizadas na Pesquisa pelo usuário do App do MaxPromotor em campo.
prow_assunto	Tem como função armazenar as informações do Assunto que será abordado na Pesquisa pelo usuário do App do MaxPromotor em campo.
prow_pesquisa	Tem como função armazenar as informações do Assunto que será abordado na Pesquisa pelo usuário do App do MaxPromotor em campo.
prow_item_avaliado	Tem como função armazenar as informações do Item Avaliado, para ser utilizado pelo usuário do App do MaxPromotor.
prow_foto	Tem como função armazenar as informações das fotos que foram criadas pelo usuário do App do MaxPromotor.
prow_empresa	Tem como função armazenar as informações das filiais ou empresas na base de dados do MaxPromotor WEB.
prow_rede	Tem como função armazenar as redes, que estão associadas aos PDVs, para que os PDVs possam estar inseridos e ter vínculos entre eles caso estejam na mesma rede.
prow_foto_resposta_pesquisa	Tem como função associar as pesquisas com as fotos criadas pelos usuários do App do MaxPromotor.
prow_foto_resposta_pesquisa_item	Tem como função associar os itens da pesquisa com as fotos criadas pelos usuários do App do MaxPromotor.
prow_objetivo	Tem como função criar metas de avaliação para os usuários do App do MaxPromotor, para que futuramente possam ver se os objetivos foram alcançados.
prow_localizacao	Tem como função armazenar informações da geolocalização dos eventos realizados pelos usuários do App do MaxPromotor.

Principais consultas de relatórios existentes

Observação

Essas consultas não devem ser repassadas para os clientes em nenhuma hipótese, serve apenas para orientar e exemplificar a obtenção de determinadas informações no sistema.

- **Select de itens do relatório - Visitados x Não Visitado.**

```
select distinct * from (select visitados.* from (select distinct
  case when visita.id is not null and visita.motivo_visita_id is not
null then 'S' else 'N' end visitado,
  case when visita.id is not null and visita.motivo_visita_id is null
then 'S' else 'N' end pesquisado,
  pdv.codigo codigoPDV,
  cast(visita.ts_entrada as date) dataVisita,
  to_char(visita.ts_entrada + interval '1h' * visita.timezone) ,
'HH24:MI:SS') checkin,
  to_char(visita.ts_saida + interval '1h' * visita.timezone) ,
'HH24:MI:SS') checkout,
  pdv.nome razaoPDV,
  pdv.fantasia fantasiaPDV,
  cast(date_part('hour', visita.TS_SAIDA - visita.TS_ENTRADA) * 60 *
60 + DATE_PART('minute', visita.TS_SAIDA - visita.TS_ENTRADA) * 60 +
DATE_PART('second', visita.TS_SAIDA - visita.TS_ENTRADA) as bigint)
tempoPermanencia,
  cast((select DATE_PART('hour', visita.TS_ENTRADA - TS_SAIDA) * 60 *
60 + DATE_PART('minute', visita.TS_ENTRADA - TS_SAIDA) * 60 +
DATE_PART('second', visita.TS_ENTRADA - TS_SAIDA) from prow_visita
where usuario_id = visita.usuario_id and date_trunc('day', ts_saida) =
date_trunc('day', visita.ts_entrada) and ts_saida < visita.ts_entrada
order by ts_entrada desc limit 1) as bigint) tempoDeslocamento,
  pdv.ts_ultima_compra dataUltimaCompra,
  (select min(data) from prow_agenda ag join prow_item_agenda it on
ag.id = it.agenda_id where ag.usuario_id = visita.usuario_id and
it.ponto_venda_id = pdv.id and it.atendida = false and data >
current_date) dataProximaVisita,
  (select count(1) from prow_visita where ponto_venda_id =
visita.ponto_venda_id and usuario_id = visita.usuario_id and ts_entrada
BETWEEN :dataInicio AND :dataFim) quantidadeVisitas,
  (select case when it.ponto_venda_id is not null then 'S' else 'N'
and
  from prow_agenda ag join prow_item_agenda it on ag.id =
it.agenda_id
  where ag.usuario_id = pdvusr.usuario_id and
it.ponto_venda_id = visita.ponto_venda_id and ag.data =
cast(visita.ts_entrada as date) limit 1) visitaAgenda,
  (select max(data) from prow_agenda ag join prow_item_agenda it on
ag.id = it.agenda_id where ag.usuario_id = visita.usuario_id and
it.ponto_venda_id = pdv.id and data < cast(visita.ts_entrada as date)
and visita.tipo_justificativa = 'VISITA AVULSA')
dataRealProgramadaVisitaAvulsa
FROM
  prow_ponto_venda pdv
inner join prow_visita visita
  on pdv.id = visita.ponto_venda_id
inner join prow_view_usuario_ponto_venda_auto_rel pdvusr
  on pdvusr.ponto_venda_id = pdv.id
  and pdvusr.usuario_id = visita.usuario_id
left join prow_visita_resposta_pesquisa visita_resp_pesquisa
  ON visita.id = visita_resp_pesquisa.visita_id
left join prow_resposta_pesquisa resposta_pesquisa
  ON resposta_pesquisa.id = visita_resp_pesquisa.resposta_pesquisa_id
where visita.usuario_id = :idPromotor
and visita.ts_entrada BETWEEN :dataInicio AND :dataFim
and ((:agendado <> '0' and pdvusr.ponto_venda_id not in (select
pit.ponto_venda_id from prow_agenda ag inner join prow_item_agenda it
on ag.id = it.agenda_id where ag.data BETWEEN :dataInicio AND
:dataFim and ag.usuario_id = pdvusr.usuario_id)
  or (:agendado <> '1' and pdvusr.ponto_venda_id in
(select it.ponto_venda_id from prow_agenda ag inner join
prow_item_agenda it on ag.id = it.agenda_id where ag.data BETWEEN
:dataInicio AND :dataFim and ag.usuario_id = pdvusr.usuario_id)))

order by pdv.nome, dataVisita) visitados
where (visitados.pesquisado = case when :pesquisado <> '1' then 'S'
else 'N' end or :pesquisado = '2' or :pesquisado = '3')
and (visitados.visitado = case when :visitado <> '1' then 'S' else 'N'
end or :visitado = '2' or :visitado = '3')

UNION

select nao_visitados.* from (
  -- nao visitados
  select distinct
    'N' visitado,
```

```

        from prow_agenda ag join prow_item_agenda it on ag.id =
it.agenda_id
        where ag.usuario_id = pdvusr.usuario_id and
it.ponto_venda_id = visita.ponto_venda_id and ag.data =
cast(visita.ts_entrada as date) limit 1) visitaAgenda,
        (select max(data) from prow_agenda ag join prow_item_agenda it on
ag.id = it.agenda_id where ag.usuario_id = visita.usuario_id and
it.ponto_venda_id = pdv.id and data < cast(visita.ts_entrada as date)
and visita.tipo_justificativa = 'VISITA AVULSA')
dataRealProgramadaVisitaAvulsa
FROM      prow_ponto_venda pdv
inner join prow_visita visita
on pdv.id = visita.ponto_venda_id
inner join prow_view_usuario_ponto_venda_auto_rel pdvusr
on pdvusr.ponto_venda_id = pdv.id
and pdvusr.usuario_id = visita.usuario_id
left join prow_visita_resposta_pesquisa visita_resp_pesquisa
ON visita.id = visita_resp_pesquisa.visita_id
left join prow_resposta_pesquisa resposta_pesquisa
ON resposta_pesquisa.id = visita_resp_pesquisa.resposta_pesquisa_id
where visita.usuario_id = :idPromotor
and visita.ts_entrada BETWEEN :dataInicio AND :dataFim
and ((:agendado <> '0' and pdvusr.ponto_venda_id not in (select
pit.ponto_venda_id from prow_agenda ag inner join prow_item_agenda it
on ag.id = it.agenda_id where ag.data BETWEEN :dataInicio AND
:dataFim and ag.usuario_id = pdvusr.usuario_id))
or (:agendado <> '1' and pdvusr.ponto_venda_id in
(select it.ponto_venda_id from prow_agenda ag inner join
prow_item_agenda it on ag.id = it.agenda_id where ag.data BETWEEN
:dataInicio AND :dataFim and ag.usuario_id = pdvusr.usuario_id)))

order by pdv.nome, dataVisita) visitados
where (visitados.pesquisado = case when :pesquisado <> '1' then 'S'
else 'N' end or :pesquisado = '2' or :pesquisado = '3')
and (visitados.visitado = case when :visitado <> '1' then 'S' else 'N'
end or :visitado = '2' or :visitado = '3')

UNION

select nao_visitados.* from (
-- nao visitados
select distinct
'N' visitado,

```

```

'N' pesquisado,
pdv.codigo_codigoPDV,
cast(null as date) dataVisita,
'-' checkin,
'-' checkout,
pdv.nome razaoPDV,
pdv.fantasia fantasiaPDV,
0 tempoPermanencia,
0 tempoDeslocamento,
pdv.ts_ultima_compra dataUltimaCompra,
(select min(data) from prow_agenda ag join prow_item_agenda
it on ag.id = it.agenda_id where ag.usuario_id = pdvusr.usuario_id and
it.ponto_venda_id = pdv.id and it.atendida = false and data >=
current_date) dataProximaVisita,
0 quantidadeVisitas,
(select case when it.ponto_venda_id is not null then 'S'
else 'N' end
from prow_agenda ag join prow_item_agenda it on ag.id =
it.agenda_id
where ag.usuario_id = pdvusr.usuario_id and
it.ponto_venda_id = pdvusr.ponto_venda_id and ag.data BETWEEN
:dataInicio AND :dataFim limit 1) visitaAgenda,
cast(null as date) dataRealProgramadaVisitaAvulsa
FROM      prow_ponto_venda pdv
inner join prow_view_usuario_ponto_venda_auto_rel pdvusr
on pdvusr.ponto_venda_id = pdv.id
where pdvusr.usuario_id = :idPromotor and
((:agendado <> '0' and pdvusr.ponto_venda_id not in (select
it.ponto_venda_id from prow_agenda ag inner join prow_item_agenda it on
ag.id = it.agenda_id where ag.data BETWEEN :dataInicio AND :dataFim
and ag.usuario_id = pdvusr.usuario_id))
or (:agendado <> '1' and pdvusr.ponto_venda_id in
(select it.ponto_venda_id from prow_agenda ag inner join
prow_item_agenda it on ag.id = it.agenda_id where ag.data BETWEEN
:dataInicio AND :dataFim and ag.usuario_id = pdvusr.usuario_id)))
and pdv.id not in (select distinct visita.ponto_venda_id from
prow_visita visita where visita.ts_entrada BETWEEN :dataInicio AND
:dataFim and visita.usuario_id = pdvusr.usuario_id)
and (:visitado <> '0' and :pesquisado <> '0')

order by pdv.nome) nao_visitados) visitas
ORDER BY visitas.razaoPDV, visitas.dataVisita

```

- Select principal do relatório - Visitados x Não Visitado.

```
SELECT supervisor.codigo codigosupervisor,
       supervisor.nome nomesupervisor,
       prow_usuario.codigo codigopromotor,
       prow_usuario.nome nomepromotor,
       (SELECT COUNT (DISTINCT pdvusr.ponto_venda_id)
        FROM   prow_visita visita
              INNER JOIN
                 prow_view_usuario_ponto_venda_auto_rel pdvusr
                ON pdvusr.ponto_venda_id = visita.ponto_venda_id
        WHERE  visita.usuario_id = prow_usuario.id
              AND pdvusr.usuario_id = prow_usuario.id
              AND ts_entrada BETWEEN :dataInicio AND :dataFim)
       quantidadepontosvendavisitado,
       quantidadepontosvenda,
       (select count(distinct ponto_venda_id) from prow_agenda ag join
prow_item_agenda it on ag.id = it.agenda_id where ag.usuario_id = |
pdvusr.usuario_id and data BETWEEN :dataInicio AND :dataFim and
ponto_venda_id in (select ponto_venda_id from
prow_view_usuario_ponto_venda_auto_rel where usuario_id =
pdvusr.usuario_id)) quantidadePontosVendaAgenda,
       (SELECT COUNT (visita.ponto_venda_id)
        FROM prow_visita visita
        WHERE visita.usuario_id = prow_usuario.id

              AND ts_entrada BETWEEN :dataInicio AND :dataFim)
       quantidadevisitas,
       (SELECT CAST (SUM (date_part ('hour',visita.ts_saida -
visita.ts_entrada) * 60 * 60 + date_part ('minute',visita.ts_saida -
visita.ts_entrada) * 60 + date_part ('second',visita.ts_saida -
visita.ts_entrada)) AS bigint)
        FROM prow_visita visita
        WHERE  visita.usuario_id = prow_usuario.id
              AND ts_entrada BETWEEN :dataInicio AND :dataFim)
       tempopermanencia,
       (SELECT CAST (SUM (date_part ('hour', fim - inicio) * 60 * 60
+ date_part ('minute', fim - inicio) * 60 + date_part ('second', fim -
inicio)) AS bigint)
        FROM (SELECT ts_entrada fim,
                    LAG (ts_saida, 1, NULL) OVER (ORDER BY
ts_saida)
```

```
         inicio,
         ponto_venda_id
        FROM prow_visita visita
        WHERE  usuario_id = prow_usuario.id
              AND ts_entrada BETWEEN :dataInicio AND
:dataFim) t
       WHERE CAST (inicio AS DATE) = CAST (fim AS DATE))
       tempodeslocamento,
       (SELECT COUNT (DISTINCT ponto_venda_id)
        FROM prow_visita visita
        WHERE  visita.usuario_id = prow_usuario.id
              AND ts_entrada BETWEEN :dataInicio AND :dataFim
              AND EXISTS
                 (SELECT 'x'
                  FROM   prow_agenda ag
                        JOIN
                           prow_item_agenda it
                          ON ag.id = it.agenda_id
                  WHERE  ponto_venda_id =
visita.ponto_venda_id
                      AND data BETWEEN :dataInicio AND
:dataFim))
       quantidadepontosvendavisitadoagenda,
       prow_usuario.id,
       perfil,
       (SELECT COUNT (tipo_justificativa)
        FROM   prow_visita visita
              INNER JOIN
                 prow_view_usuario_ponto_venda_auto_rel
pdvusr
                ON pdvusr.ponto_venda_id =
visita.ponto_venda_id
        WHERE  visita.usuario_id = prow_usuario.id
              AND pdvusr.usuario_id = prow_usuario.id
              AND ts_entrada BETWEEN :dataInicio AND
:dataFim
              AND visita.tipo_justificativa is not null)
       visita_justificada,
       (SELECT count (pdv.pontoVenda) from (SELECT visita.ponto_venda_id as
pontoVenda , date(ts_entrada)
        FROM prow_visita visita
        WHERE visita.usuario_id = prow_usuario.id
              AND ts_entrada BETWEEN :dataInicio AND :dataFim
```

```

AND exists(SELECT ponto_venda_id as pontoVenda, data
           FROM prow_agenda ag
           JOIN prow_item_agenda it ON ag.id
= it.agenda_id
           WHERE ag.usuario_id =
pdvusr.usuario_id
           AND visita.ponto_venda_id =
it.ponto_venda_id
           AND date(visita.ts_entrada) =
ag.data
           AND DATA BETWEEN :dataInicio
AND :dataFim
           AND ponto_venda_id IN
           (SELECT ponto_venda_id FROM
prow_view_usuario_ponto_venda_auto_rel WHERE usuario_id =
pdvusr.usuario_id)
           group by ponto_venda_id, data) group by ponto_venda_id,
date(ts_entrada)) as pdv) quantidadePDVsVisitadoDentroDaAgenda,
(SELECT COUNT(pdv.pontoVenda) from (SELECT visita.ponto_venda_id as
pontoVenda , date(ts_entrada)
FROM prow_visita visita
WHERE visita.usuario_id = prow_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
AND not exists(SELECT ponto_venda_id as pontoVenda, data
                FROM prow_agenda ag
                JOIN prow_item_agenda it ON ag.id
= it.agenda_id
                WHERE ag.usuario_id =
pdvusr.usuario_id
                AND visita.ponto_venda_id =
it.ponto_venda_id
                AND date(visita.ts_entrada) =
ag.data
                AND DATA BETWEEN :dataInicio
AND :dataFim
                AND ponto_venda_id IN
                (SELECT ponto_venda_id
FROM
prow_view_usuario_ponto_venda_auto_rel
                WHERE usuario_id =
pdvusr.usuario_id) group by ponto_venda_id, data) group by
ponto_venda_id, date(ts_entrada)) as pdv)
quantidadePDVsVisitadosForaRota,

```

```

(SELECT COUNT (pdv.pontoVenda) from (SELECT it.ponto_venda_id as
pontoVenda, data
FROM prow_agenda ag
JOIN prow_item_agenda it ON ag.id = it.agenda_id
WHERE ag.usuario_id = pdvusr.usuario_id
AND DATA BETWEEN :dataInicio AND :dataFim
AND ponto_venda_id in
(select ponto_venda_id from
prow_view_usuario_ponto_venda_auto_rel where usuario_id =
pdvusr.usuario_id)
group by ponto_venda_id, data) as pdv)
quantidadePDVsProgramadoAgenda
FROM prow_view_usuario_ponto_venda_auto_rel pdvusr
INNER JOIN prow_usuario
ON prow_usuario.id = usuario_id
join prow_usuario.perfil perfil
on prow_usuario.perfil_id = perfil.id
LEFT OUTER JOIN ( SELECT COUNT (DISTINCT pdv.ponto_venda_id)
quantidadePontosVenda,
usuario_id uid_qtd
FROM prow_view_usuario_ponto_venda_auto_rel
pdv
GROUP BY usuario_id) pdvqtd
ON (pdvqtd.uid_qtd = prow_usuario.id)
LEFT OUTER JOIN (SELECT usuario_supervisor.codigo,
usuario_supervisor.nome,
subordinado.subordinado_id,
subordinado.superior_id
FROM prow_usuario usuario_supervisor
INNER JOIN
prow_usuario_subordinado subordinado
ON usuario_supervisor.id =
subordinado.superior_id) supervisor
ON (supervisor.subordinado_id = prow_usuario.id)
WHERE
1 = 1

AND perfil in ('PRO', 'VEN')
GROUP BY supervisor.codigo, supervisor.nome, prow_usuario.codigo,
prow_usuario.nome, prow_usuario.id, quantidadePontosVenda,
pdvusr.usuario_id, perfil
ORDER BY supervisor.nome, prow_usuario.codigo, prow_usuario.nome

```

- Select relatório - Apuração.

```

SELECT agrupamento,
       nome_pesquisa,
       nome_pdv,
       nome_fantasia_pdv,
       cidade,
       promotor,
       supervisor,
       vendedor,
       MAX(data_hora_checkin) AS data_hora_checkin,
       MAX(data_hora_checkout) AS data_hora_checkout,
       item_avaliado,
       pergunta,
       valor_resposta,
       tipo_pergunta_resposta,
       foto,
       MAX(ts_entrada) AS ts_entrada,
       pesquisa,
       resposta,
       id_item_agrupado,
       id_resposta_agrupada,
       sequencia_resposta,
       assunto,
       sequencia_assunto,
       item_avaliado_id,
       ponto_venda_id,
       pesquisa_id
FROM
  (SELECT CASE
        WHEN :tipoAgrupamento = 1 THEN pesquisa.descricao
        WHEN :tipoAgrupamento = 2 THEN
resposta.item_avaliado_nome
        WHEN :tipoAgrupamento = 3 THEN
        (SELECT usuario_supervisor.nome
         FROM prow_usuario usuario_supervisor
         INNER JOIN prow_usuario_subordinado subordinado
ON usuario_supervisor.id = subordinado.superior_id
         WHERE subordinado.subordinado_id =
usuario_promotor.id
         LIMIT 1)
        WHEN :tipoAgrupamento = 4 THEN usuario_promotor.nome
        WHEN :tipoAgrupamento = 5 THEN pdv.nome

```

```

        WHEN :tipoAgrupamento = 6 THEN
        (SELECT cidade.nome
         FROM PROW_ENDEREÇO_PONTO_VENDA
endereco_ponto_venda
         JOIN PROW_ENDEREÇO endereco ON
endereco_ponto_venda.endereco_id = endereco.id
         AND endereco_ponto_venda.ponto_venda_id = pdv.id
         JOIN PROW_CIDADE cidade ON endereco.cidade_id =
cidade.id
         ORDER BY endereco.principal
         LIMIT 1)
        END AS agrupamento,
       pesquisa.descricao nome_pesquisa,
       pdv.nome nome_pdv,
       pdv.fantasia nome_fantasia_pdv,
       pdv.id ponto_venda_id,

  (SELECT cidade.nome
   FROM PROW_ENDEREÇO_PONTO_VENDA endereco_ponto_venda
   JOIN PROW_ENDEREÇO endereco ON endereco_ponto_venda.endereco_id =
endereco.id
   AND endereco_ponto_venda.ponto_venda_id = pdv.id
   JOIN PROW_CIDADE cidade ON endereco.cidade_id = cidade.id
   ORDER BY endereco.principal
   LIMIT 1) AS cidade,
       usuario_promotor.nome promotor,

  (SELECT usuario_supervisor.nome
   FROM prow_usuario usuario_supervisor
   INNER JOIN prow_usuario_subordinado subordinado ON
usuario_supervisor.id = subordinado.superior_id
   WHERE subordinado.subordinado_id = usuario_promotor.id
   LIMIT 1) AS supervisor,

  (SELECT usuario_vendedor.nome
   FROM prow_usuario usuario_vendedor
   INNER JOIN prow_usuario_ponto_venda usuario_ponto_venda ON
usuario_vendedor.id = usuario_ponto_venda.usuario_id
   INNER JOIN prow_usuario_perfil usuario_perfil ON
usuario_vendedor.perfil_id = usuario_perfil.id
   WHERE usuario_perfil.perfil = 'VEN'
   AND pdv.id = usuario_ponto_venda.ponto_venda_id
   LIMIT 1) AS vendedor,

```



```

        visita.ts_entrada + interval '1h' * visita.timezone
data_hora_checkin,
        visita.ts_saida + interval '1h'
* visita.timezone data_hora_checkout,
resposta.item_avaliado_nome item_avaliado,
resposta.item_avaliado_id item_avaliado_id,
resposta.pergunta_descricao pergunta,
CASE
WHEN (resposta.justificado_checkout = TRUE) THEN 'JUSTIFICADO CHECKOUT'
ELSE resposta.item_valor_resposta
END
valor_resposta,
resposta.pergunta_tipo tipo_pergunta_resposta,
resposta.foto,
resposta.tipo_foto,
resposta.data_foto,
resposta_pesquisa_item_id,
visita.ts_entrada,
pesquisa.id pesquisa,
resposta_pesquisa.id resposta,
resposta.id item_agrupado,
resposta.id resposta_agrupada,
resposta.sequencia_pergunta,
resposta.sequencia_resposta,
assunto_pesquisa.sequencia_sequencia_assunto,

```

```

assunto_descricao assunto,
resposta_pesquisa.pesquisa_id as pesquisa_id
FROM prow_visita visita
INNER JOIN prow_visita_resposta_pesquisa visita_resp_pesquisa ON
visita.id = visita_resp_pesquisa.visita_id
INNER JOIN prow_resposta_pesquisa resposta_pesquisa ON
resposta_pesquisa.id = visita_resp_pesquisa.resposta_pesquisa_id
INNER JOIN prow_pesquisa pesquisa ON resposta_pesquisa.pesquisa_id =
pesquisa.id
INNER JOIN prow_ponto_venda pdv ON pdv.id = visita.ponto_venda_id
INNER JOIN
(SELECT pergunta_descricao,
item_avaliado_id,
item_avaliado_nome,
item_valor_resposta,
resposta_pesquisa_id,
0 id_item_agrupado,
0 id_resposta_agrupada,
0 sequencia_pergunta,
0 sequencia_resposta,
foto.caminho_foto,
foto_resp_pesq_item.tipo tipo_foto,
(foto.ts_captura + interval '1h' * foto.timezone)
data_foto,
prow_view_resposta.resposta_pesquisa_item_id,
null resposta_pesquisa_item_agrup_id,
justificado_checkout,
pergunta_tipo,
perfil
FROM prow_view_resposta
JOIN
(SELECT valor_configuracao_usuario(prow_usuario.id,
'CONFIG_EXIBIR_INFORMACAO_FOTO') info_foto,
prow_usuario.*
FROM prow_usuario) usuario_promotor ON
prow_view_resposta.usuario_id = usuario_promotor.id
JOIN prow_usuario_perfil perfil ON usuario_promotor.perfil_id =
perfil.id
LEFT JOIN prow_foto_resp_pesq_item foto_resp_pesq_item ON
prow_view_resposta.resposta_pesquisa_item_id =
foto_resp_pesq_item.resposta_pesquisa_item_id

```

```

LEFT JOIN prow_foto foto ON (usuario_promotor.info_foto = 'false'
AND foto_resp_pesq_item.foto_original_id = foto.id) OR (NOT
usuario_promotor.info_foto = 'false' AND
foto_resp_pesq_item.foto_marca_id = foto.id)
WHERE :tipoPergunta IN ('TODAS', 'SIMPLES')
AND data_visita BETWEEN :dataInicio AND :dataFim

AND item_valor_resposta IS NOT NULL
AND item_valor_resposta <> ''
UNION
SELECT pergunta_descricao,
       item_avaliado_id,
       item_avaliado_nome,
       item_valor_resposta,
       resposta_pesquisa_id,
       item_avali_perg_agrup_pesq_id id item_agrupado,

prow_view_resposta_agrupada.resposta_pesquisa_item_agrup_id
id resposta_agrupada,
       sequencia_pergunta,
       sequencia_resposta,
       foto.caminho_foto,
       foto_resp_pesq_item.tipo tipo_foto,
       (foto.ta_captura + interval '1h' * foto.timezone)

data_foto,
       null resposta_pesquisa_item_id,

prow_view_resposta_agrupada.resposta_pesquisa_item_agrup_id,
       justificado_checkout,
       pergunta_tipo,
       perfil

FROM prow_view_resposta_agrupada
JOIN
       (SELECT valor_configuracao_usuario(prow_usuario.id,
'CONFIG_EXIBIR_INFORMACAO_FOTO') info_foto,
       prow_usuario.*
FROM prow_usuario) usuario_promotor ON
prow_view_resposta_agrupada.usuario_id = usuario_promotor.id
JOIN prow_usuario_perfil perfil ON usuario_promotor.perfil_id =
perfil.id
LEFT JOIN prow_foto_resp_pesq_item_agrup foto_resp_pesq_item ON
prow_view_resposta_agrupada.resposta_pesquisa_item_agrup_id =
foto_resp_pesq_item.resposta_pesquisa_item_agrup_id

```

```

LEFT JOIN prow_foto foto ON (usuario_promotor.info_foto = 'false'
AND foto_resp_pesq_item.foto_original_id = foto.id) OR ((NOT
usuario_promotor.info_foto = 'false') AND
foto_resp_pesq_item.foto_marca_id = foto.id)
WHERE :tipoPergunta IN ('TODAS', 'AGRUPADA')
AND data_visita BETWEEN :dataInicio AND :dataFim

AND item_valor_resposta IS NOT NULL
AND item_valor_resposta <> ''
) resposta ON resposta.resposta_pesquisa_id = resposta_pesquisa.id
LEFT JOIN prow_resposta_pesquisa_item resposta_pesquisa_item ON
resposta_pesquisa_item.id = resposta.resposta_pesquisa_item_id
LEFT JOIN prow_item_avali_perg_pesq_item_avali_perg_pesq ON
item_avali_perg_pesq.id =
resposta_pesquisa_item.item_avali_perg_pesq_id
LEFT JOIN prow_perg_assun_pesq_perg_assun_pesq ON perg_assun_pesq.id
= item_avali_perg_pesq.pergunta_assun_pesq_id

LEFT JOIN prow_resposta_pesquisa_item_agrup
resposta_pesquisa_item_agrup ON resposta_pesquisa_item_agrup.id =
resposta.resposta_pesquisa_item_agrup_id
LEFT JOIN prow_resposta_pesquisa_agrup resposta_pesquisa_agrup ON
resposta_pesquisa_agrup.id =
resposta_pesquisa_item_agrup.resposta_pesquisa_agrup_id
LEFT JOIN prow_item_avali_perg_agrup_pesq_item_avali_perg_agrup_pesq
ON item_avali_perg_agrup_pesq.id =
resposta_pesquisa_agrup.item_avali_perg_agrup_pesq_id
LEFT JOIN prow_perg_agrup_assun_pesq_perg_agrup_assun_pesq ON
perg_agrup_assun_pesq.id =
item_avali_perg_agrup_pesq.pergunta_assun_agrup_pesq_id

LEFT JOIN prow_assunto_pesquisa assunto_pesquisa ON
(assunto_pesquisa.id = perg_assun_pesq.assunto_pesquisa_id OR
assunto_pesquisa.id = perg_agrup_assun_pesq.assunto_pesquisa_id)
LEFT JOIN prow_assunto_assunto ON assunto.id =
assunto_pesquisa.id assunto
LEFT JOIN prow_usuario usuario_promotor ON visita.usuario_id =
usuario_promotor.id
WHERE visita.ts_entrada BETWEEN :dataInicio AND :dataFim

AND perfil in ('PRO', 'VEN')
AND pesquisa.id IN (:pesquisas)

```

```

) t
GROUP BY agrupamento,
    nome_pesquisa,
    nome_rdv,
    nome_fantasia_rdv,
    cidade,
    promotor,
    supervisor,
    vendedor,

    item_avaliado,
    pergunta,
    valor_resposta,
    tipo_pergunta_resposta,
    foto,

    pesquisa,
    resposta,
    id_item_agrupado,
    id_resposta_agrupada,
    sequencia_pergunta,
    sequencia_resposta,
    tipo_foto,
    data_foto,
    resposta_pesquisa_item_id,
    sequencia_assunto,
    assunto,
    item_avaliado_id,
    ponto_venda_id,
    pesquisa_id
ORDER BY agrupamento,
    nome_rdv,
    ta_entrada DESC,
    id_item_agrupado,
    sequencia_assunto,
    sequencia_pergunta,
    sequencia_resposta,
    resposta_pesquisa_item_id,
    data_foto,
    tipo_foto

```

- Select relatório - Objetivo de Metas.

```

select
    codigoSupervisor,
    nomeSupervisor,
    codigoPromotor,
    nomePromotor,
    objetivo,
    meta_min,
    meta_max,
    case when tipoPremio = 'REAIS'
        then cast('R$ ' || cast(premio_min as numeric(20, 2)) as text)
    when tipoPremio = 'PERCENTUAL'
        then cast(cast(premio_min as numeric(20, 2)) || '%' as text)
    else '0' end as premio_minimo,
    case when tipoPremio = 'REAIS'
        then cast('R$ ' || cast(premio_max as numeric(20, 2)) as text)
    when tipoPremio = 'PERCENTUAL'
        then cast(cast(premio_max as numeric(20, 2)) || '%' as text)
    else '0' end as premio_maximo,
    atingido,
    case when tipoPremio = 'REAIS'
        then cast('R$ ' || COALESCE((select premio from
    from_prova_valor_objetivo_usuario where objetivo_usuario_id = ou_id and
    (cast(atingido as numeric(20, 2)) >= valor) order by valor desc limit
    1), 0) as text)
    when tipoPremio = 'PERCENTUAL'
        then cast(COALESCE((select cast(atingido * (premio / 100) as
    int) from from_prova_valor_objetivo_usuario where objetivo_usuario_id =
    ou_id and (cast(atingido as numeric(20, 2)) >= valor) order by valor
    desc limit 1), 0) || '%' as text)
    else '0' end as premio_atingido,
    tipo,
    idObjItemAvaliado,
    nomeItemAvaliado,
    case when atingido <> 0 and atingido is not null then
        cast(((atingido*100)/meta_max) as numeric(20, 2)) else '0' end as
    percentualValorAtingido
    from
    (select
        supervisor.codigo_codigoSupervisor,
        supervisor.nome_nomeSupervisor,
        promotor.codigo_codigoPromotor,

```

```

promotor.nome nomePromotor,
objetivo.descricao objetivo,
objetivo.tipo premio tipoPremio,
objetivo.usuario_id as ou_id,
(select valor from prow_faixa_valor_objetivo_usuario where
objetivo_usuario_id = objetivo_usuario.id order by valor asc limit 1)
meta_min,
(select valor from
prow_faixa_valor_objetivo_usuario where objetivo_usuario_id =
objetivo_usuario.id order by valor desc limit 1) meta_max,
(select premio from
prow_faixa_valor_objetivo_usuario where objetivo_usuario_id =
objetivo_usuario.id order by valor asc limit 1) premio_min,
(select premio from
prow_faixa_valor_objetivo_usuario where objetivo_usuario_id =
objetivo_usuario.id order by valor desc limit 1) premio_max,
case
when objetivo.tipo = 'QUANTIDADE_VISITA' then
(select count(pdv.pontoVenda) from
(select visita.ponto_venda_id as pontoVenda ,
date(ts_entrada)
from prow_visita visita
where visita.usuario_id = promotor.id
and ts_entrada between :dataInicio and
:dataFim
group by ponto_venda_id, date(ts_entrada)) as
pdv
)
when objetivo.tipo = 'QUANTIDADE_FOTO' then
(select sum(total) from
(
select count(distinct
FROM FOTO_RESPONSA_PESQUISA.id) total from prow_visita visita
inner join PROW_USUARIO_PONTO_VENDA pdvusr on
pdvusr.ponto_venda_id = visita.ponto_venda_id
inner join prow_visita_resposta_pesquisa ON
prow_visita_resposta_pesquisa.visita_id = visita.id
inner join PROW_FOTO_RESPONSA_PESQUISA ON
PROW_FOTO_RESPONSA_PESQUISA.resposta_pesquisa_id =
prow_visita_resposta_pesquisa.resposta_pesquisa_id
where visita.usuario_id = promotor.id
and visita.ts_entrada BETWEEN :dataInicio AND
:dataFim
)
)
union all
select count(distinct FROM FOTO_RESP_PESQ_ITEM.id)
total from prow_visita visita
inner join PROW_USUARIO_PONTO_VENDA pdvusr on
pdvusr.ponto_venda_id = visita.ponto_venda_id
inner join prow_visita_resposta_pesquisa ON
prow_visita_resposta_pesquisa.visita_id = visita.id
inner join prow_resposta_pesquisa_item ON
prow_resposta_pesquisa_item.resposta_pesquisa_id =
prow_visita_resposta_pesquisa.resposta_pesquisa_id
inner join PROW_FOTO_RESP_PESQ_ITEM on
PROW_FOTO_RESP_PESQ_ITEM.resposta_pesquisa_item_id =
prow_resposta_pesquisa_item.id
where visita.usuario_id = promotor.id
and visita.ts_entrada BETWEEN :dataInicio AND
:dataFim
) t(foto)
when objetivo.tipo = 'QUANTIDADE_JUSTIFICATIVA' then
(select count(distinct pdvusr.ponto_venda_id) from
prow_visita visita
inner join PROW_USUARIO_PONTO_VENDA pdvusr on
pdvusr.ponto_venda_id = visita.ponto_venda_id
where visita.usuario_id = promotor.id
and visita.tipo_justificativa = 'NAO_VISITA'
and ts_entrada BETWEEN :dataInicio AND :dataFim
)
when objetivo.tipo = 'VALOR_VENDIDO' then
(select COALESCE(sum(valor_unitario * qtd_itens),
0) from prow_pedido_produto
inner join prow_pedido on prow_pedido.id =
prow_pedido_produto.pedido_id
inner join prow_item_avaliado on
prow_item_avaliado.id = prow_pedido_produto.item_avaliado_id
where ts_data_pedido BETWEEN :dataInicio AND
:dataFim
and ponto_venda_id in (select
visita.ponto_venda_id from prow_visita visita
inner join
PROW_USUARIO_PONTO_VENDA pdvusr on pdvusr.ponto_venda_id =
visita.ponto_venda_id
where visita.usuario_id
= promotor.id

```

```

)
when objetivo.tipo = 'QUANTIDADE_JUSTIFICATIVA' then
(select count(distinct pdvusr.ponto_venda_id) from
prow_visita visita
inner join PROW_USUARIO_PONTO_VENDA pdvusr on
pdvusr.ponto_venda_id = visita.ponto_venda_id
where visita.usuario_id = promotor.id
and visita.tipo_justificativa = 'NAO_VISITA'
and ts_entrada BETWEEN :dataInicio AND :dataFim
)
when objetivo.tipo = 'VALOR_VENDIDO' then
(select COALESCE(sum(valor_unitario * qtd_itens),
0) from prow_pedido_produto
inner join prow_pedido on prow_pedido.id =
prow_pedido_produto.pedido_id
inner join prow_item_avaliado on
prow_item_avaliado.id = prow_pedido_produto.item_avaliado_id
where ts_data_pedido BETWEEN :dataInicio AND
:dataFim
and ponto_venda_id in (select
visita.ponto_venda_id from prow_visita visita
inner join
PROW_USUARIO_PONTO_VENDA pdvusr on pdvusr.ponto_venda_id =
visita.ponto_venda_id
where visita.usuario_id
= promotor.id

```

```

                                and ts_entrada BETWEEN
:dataInicio AND :dataFim)
                                and prow_item_avaliado.fornecedor_id =
obj_item_avaliado.item_avaliado_id
                                )
                                when objetivo.tipo = 'MIX_VENDIDO' then
                                (select count(prow_mix_cliente.id) from
prow_mix_cliente
                                inner join prow_item_avaliado on
prow_item_avaliado.id = prow_mix_cliente.item_avaliado_id
                                where ts_data_venda_item_aval BETWEEN
:dataInicio AND :dataFim
                                and ponto_venda_id in (select
visita.ponto_venda_id from prow_visita visita
                                inner join
PROW_USUARIO_PONTO_VENDA pdvusr on pdvusr.ponto_venda_id =
visita.ponto_venda_id
                                where visita.usuario_id
= promotor.id
                                and ts_entrada BETWEEN
:dataInicio AND :dataFim)
                                and prow_item_avaliado.fornecedor_id =
obj_item_avaliado.item_avaliado_id
                                )
                                when objetivo.tipo = 'PERGUNTA' then
                                COALESCE((select sum(cast(item_valor_resposta as
numeric)) from
                                (
                                select visita.item_valor_id,
visita.item_valor_resposta from prow_view_resposta visita
                                inner join PROW_USUARIO_PONTO_VENDA pdvusr on
pdvusr.ponto_venda_id = visita.ponto_venda_id
                                where visita.pergunta_id = objetivo.pergunta_id
                                and visita.item_valor_resposta ~ '^[0-9\.]+$',
                                and visita.usuario_id = promotor.id
                                and visita.data_visita BETWEEN :dataInicio AND
:dataFim
                                group by visita.item_valor_id,
visita.item_valor_resposta
                                ) tpergunta), 0)
                                else
                                0 end atingido,
                                objetivo.tipo,

```

```

                                obj_item_avaliado.item_avaliado_id idObjItemAvaliado,
                                (select nome from prow_item_avaliado where id =
obj_item_avaliado.item_avaliado_id) nomeItemAvaliado
                                from prow_objetivo objetivo
                                left join prow_objetivo_item_avaliado obj_item_avaliado on
objetivo.id = obj_item_avaliado.objetivo_id
                                inner join prow_objetivo_usuario objetivo_usuario on
objetivo_usuario.objetivo_id = objetivo.id
                                inner join PROW_USUARIO promotor on objetivo_usuario.usuario_id
= promotor.id
                                left join prow_usuario_subordinado on
PROW_USUARIO_SUBORDINADO.subordinado_id = promotor.id
                                left join PROW_USUARIO supervisor on
PROW_USUARIO_SUBORDINADO.superior_id = supervisor.id
                                join prow_usuario_perfil perfil on promotor.perfil_id =
perfil.id
                                where 1 = 1

AND perfil in ('PRO', 'VEN')
                                order by supervisor.codigo, promotor.codigo,
objetivo_usuario.objetivo_id
                                ) t

```

- Select relatório - Produtividade e ociosidade.

```

SELECT supervisor.codigo codigosupervisor,
       supervisor.nome nomesupervisor,
       prow_usuario.codigo codigopromotor,
       prow_usuario.nome nomepromotor,

(SELECT MAX(pontoVenda) FROM (SELECT pontoVenda from (SELECT COUNT
(distinct (it.ponto_venda_id)) as pontoVenda
FROM prow_agenda ag
JOIN prow_item_agenda it ON ag.id = it.agenda_id
WHERE ag.usuario_id = pdvusr.usuario_id
AND DATA BETWEEN :dataInicio AND :dataFim
AND ponto_venda_id in (select ponto_venda_id from
prow_view_usuario_ponto_venda_auto_rel where usuario_id =
pdvusr.usuario_id)
group by it.ponto_venda_id, to_date(cast(it.rota_ts_data as
TEXT), 'YYYY-MM-DD')) as pdv
group by pdv.pontoVenda) valormaximoquantidadePontosVendaAgenda)
quantidadePontosVendaAgenda,

(SELECT MAX(pontoVenda) FROM (SELECT SUM(pontoVenda) pontoVenda
from (SELECT COUNT(visita.ponto_venda_id) as pontoVenda
FROM prow_visita visita
WHERE visita.usuario_id = prow_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
AND visita.ponto_venda_id IN (
(SELECT DISTINCT ponto_venda_id
FROM prow_agenda ag
JOIN prow_item_agenda it ON ag.id
= it.agenda_id
WHERE ag.usuario_id =
pdvusr.usuario_id
AND DATA BETWEEN :dataInicio
AND :dataFim
AND ponto_venda_id IN
(SELECT ponto_venda_id
FROM
prow_view_usuario_ponto_venda_auto_rel
WHERE usuario_id =
pdvusr.usuario_id))) group by ponto_venda_id, date(ts_entrada)) as pdv
group by pdv.pontoVenda) valormaximoquantidadevisitas)
quantidadevisitas,

```

```

(SELECT MAX(pontoVenda) FROM (SELECT COUNT(pontoVenda) pontoVenda from
(SELECT ponto_venda_id as pontoVenda, to_date(cast(ts_entrada as
TEXT), 'YYYY-MM-DD') FROM prow_visita visita
WHERE visita.usuario_id = prow_usuario.id
AND tipo_justificativa is null
AND visita.ts_entrada BETWEEN :dataInicio AND :dataFim
AND visita.ponto_venda_id NOT IN (
(SELECT DISTINCT
ponto_venda_id
FROM prow_agenda ag
JOIN prow_item_agenda it ON
ag.id = it.agenda_id
WHERE ag.usuario_id =
pdvusr.usuario_id
AND DATA BETWEEN
:dataInicio AND :dataFim
AND ponto_venda_id IN
(SELECT ponto_venda_id
FROM
prow_view_usuario_ponto_venda_auto_rel
WHERE usuario_id =
pdvusr.usuario_id)))
AND ts_entrada BETWEEN :dataInicio AND :dataFim
group by ponto_venda_id, to_date(cast(ts_entrada as
TEXT), 'YYYY-MM-DD')) as pdv
group by pdv.pontoVenda) valormaximoquantidadevisitasforarota)
quantidadevisitasforarota,

(SELECT CAST (SUM (date_part ('hour', visita.ts_saida -
visita.ts_entrada) * 60 + date_part ('minute', visita.ts_saida -
visita.ts_entrada) * 60 + date_part ('second', visita.ts_saida -
visita.ts_entrada)) AS bigint)
FROM prow_visita visita
WHERE visita.usuario_id = prow_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim) tempopermanencia
,

(SELECT CAST (SUM (date_part ('hour', fim - inicio) * 60 + 60 +
date_part ('minute', fim - inicio) * 60 + date_part ('second', fim -
inicio)) AS bigint)
FROM
(SELECT ts_entrada fim,

```

```

LAG (ts_saida, 1, NULL) OVER (
                                ORDER BY ts_saida) inicio,
                                ponto_venda_id
FROM provw_visita visita
WHERE usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim) t
WHERE CAST (inicio AS DATE) = CAST (fim AS DATE)) tempodealocamento
,
    pdvusr.usuario_id AS id_usuario ,
    perfil ,
    CASE WHEN provw_empresa.nome is null THEN 'SEM FILIAL' ELSE
provw_empresa.nome END AS nome_empresa ,
    provw_empresa.id AS id_empresa ,

(SELECT visita.ts_entrada
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
ORDER BY ts_entrada ASC
LIMIT 1) AS primeiro_check_in ,

(SELECT visita.ts_saida
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
ORDER BY ts_entrada DESC
LIMIT 1) AS ultimo_check_out ,
1 AS ROW ,
(CASE
    WHEN pdvusr.usuario_id IN
        (SELECT ag.usuario_id
         FROM provw_agenda ag
         JOIN provw_item_agenda it ON ag.id = it.agenda_id
         WHERE ag.usuario_id = pdvusr.usuario_id
         AND DATA BETWEEN :dataInicio AND :dataFim
         AND ponto_venda_id IN
             (SELECT ponto_venda_id
              FROM provw_view_usuario_ponto_venda_auto_rel
              WHERE usuario_id = ag.usuario_id)) THEN 1
    ELSE 0
END) AS promotor_rotreiro ,

```

```

(SELECT MAX(pontoVenda) FROM (SELECT SUM(pontoVenda) pontoVenda from
(SELECT count (visita.ponto_venda_id) as pontoVenda
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
AND visita.ponto_venda_id IN (
                                (SELECT DISTINCT ponto_venda_id
                                 FROM provw_agenda ag
                                 JOIN provw_item_agenda it ON ag.id
= it.agenda_id
                                WHERE ag.usuario_id =
pdvusr.usuario_id
                                AND DATA BETWEEN :dataInicio
AND :dataFim
                                AND ponto_venda_id IN
                                    (SELECT ponto_venda_id
                                     FROM
provw_view_usuario_ponto_venda_auto_rel
                                     WHERE usuario_id =
pdvusr.usuario_id)))
group by visita.ponto_venda_id, to_date(cast(ts_entrada as
TEXT), 'YYYY-MM-DD')) as pdv
group by pdv.pontoVenda) somaquantidadevisitasagenda)
quantidadevisitasagenda,

(SELECT COUNT(1)
FROM provw_notificacao_evento e1
WHERE ts_evento BETWEEN :dataInicio AND :dataFim
AND tipo_evento = 'CHECK_IN'
AND usuario_id = provw_usuario.id
AND ponto_venda_id NOT IN
    (SELECT distinct ponto_venda_id
     FROM provw_notificacao_evento
     WHERE ts_evento BETWEEN :dataInicio AND :dataFim
     AND (tipo_evento = 'CHECK_OUT' or tipo_evento =
'CHECK_OUT AUTOMATICO')
     AND usuario_id = e1.usuario_id
     AND ponto_venda_id = e1.ponto_venda_id)
AND ponto_venda_id NOT IN
    (SELECT distinct ponto_venda_id
     FROM provw_visita
     WHERE ts_entrada > :dataInicio and ts_saida < :dataFim
     AND usuario_id = provw_usuario.id

```

```

AND ponto_venda_id = al.ponto_venda_id) amendamentos ,
(CASE
    WHEN
        (SELECT COUNT (DISTINCT visita.ponto_venda_id)
         FROM provw_visita visita
         WHERE visita.usuario_id = provw_usuario.id
         AND visita.ponto_venda_id NOT IN (
             (SELECT
DISTINCT ponto_venda_id
FROM
provw_agenda ag
JOIN
provw_item_agenda it ON ag.id = it.agenda_id
WHERE
ag.usuario_id = pdvusr.usuario_id
AND DATA
BETWEEN :dataInicio AND :dataFim
AND
ponto_venda_id IN
(SELECT
ponto_venda_id
FROM
provw_view_usuario_ponto_venda_auto_rel
WHERE
usuario_id = pdvusr.usuario_id)))
AND ts_entrada BETWEEN :dataInicio AND :dataFim)
> 0 THEN 1
ELSE 0
END) AS promotor_fora_rota ,
COUNT(DISTINCT pdv.id) quantidade_pontos_venda,
COUNT(DISTINCT provw_usuario.id) quantidade_usuario,
(SELECT COUNT (visita.ponto_venda_id)
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND tipo_justificativa is not null
AND ts_entrada BETWEEN :dataInicio AND :dataFim)
quantidadevisitasjustificadas,
(SELECT COUNT ( DISTINCT date_part('day', visita.ts_entrada) )
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim)
quantidade_dias_trabalhados,

```

```

(SELECT COUNT (pdv.pontoVenda) from (SELECT it.ponto_venda_id aa,
pontoVenda, data
FROM provw_agenda ag
JOIN provw_item_agenda it ON ag.id = it.agenda_id
WHERE ag.usuario_id = pdvusr.usuario_id
AND DATA BETWEEN :dataInicio AND :dataFim
AND ponto_venda_id in
(select ponto_venda_id from provw_view_usuario_ponto_venda_auto_rel
where usuario_id = pdvusr.usuario_id)
group by ponto_venda_id, data) as pdv)
quantidadePDVsProgramadoAgenda,
(SELECT count(pdv.pontoVenda) from (SELECT visita.ponto_venda_id as
pontoVenda , date(ts_entrada)
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
AND exists(SELECT ponto_venda_id as pontoVenda, data
FROM provw_agenda ag
JOIN provw_item_agenda it ON ag.id
= it.agenda_id
WHERE ag.usuario_id =
pdvusr.usuario_id
AND visita.ponto_venda_id = it.ponto_venda_id
AND date(visita.ts_entrada) = ag.data
AND DATA BETWEEN :dataInicio
AND :dataFim
AND ponto_venda_id IN
(SELECT ponto_venda_id FROM
provw_view_usuario_ponto_venda_auto_rel WHERE usuario_id =
pdvusr.usuario_id)
group by ponto_venda_id, data) group by ponto_venda_id,
date(ts_entrada)) as pdv) quantidadePDVsVisitadoDentroDaAgenda,
(SELECT COUNT(pdv.pontoVenda) from (SELECT visita.ponto_venda_id as
pontoVenda , date(ts_entrada)
FROM provw_visita visita
WHERE visita.usuario_id = provw_usuario.id
AND ts_entrada BETWEEN :dataInicio AND :dataFim
AND not exists(SELECT ponto_venda_id as pontoVenda, data
FROM provw_agenda ag
JOIN provw_item_agenda it ON ag.id
= it.agenda_id
WHERE ag.usuario_id =
pdvusr.usuario_id

```



```

AND visita.ponto_venda_id = it.ponto_venda_id
AND date(visita.ts_entrada) = ag.data
AND DATA BETWEEN :dataInicio
AND :dataFim
AND ponto_venda_id IN
(SELECT ponto_venda_id
FROM
prow_view_usuario_ponto_venda_auto_rel
WHERE usuario_id =
pdvusr.usuario_id) group by ponto_venda_id, data) group by
ponto_venda_id, date(ts_entrada)) as pdv)
quantidadeEdvsVisitadosForaRota
FROM prow_view_usuario_ponto_venda_auto_rel pdvusr
INNER JOIN prow_usuario ON prow_usuario.id = usuario_id
INNER JOIN prow_ponto_venda pdv ON pdv.id = pdvusr.ponto_venda_id
LEFT JOIN prow_empresa_usuario ON prow_empresa_usuario.usuario_id =
prow_usuario.id
LEFT JOIN prow_empresa ON prow_empresa.id =
prow_empresa_usuario.empresa_id
LEFT JOIN prow_regional_empresa regional_empresa ON
regional_empresa.empresa_id = prow_empresa.id
LEFT JOIN prow_regional regional ON regional.id =
regional_empresa.regional_id
LEFT JOIN prow_usuario subordinado subordinado_supervisor ON
prow_usuario.id = subordinado_supervisor.subordinado_id
LEFT JOIN prow_usuario supervisor ON supervisor.id =
subordinado_supervisor.superior_id
LEFT JOIN prow_usuario subordinado subordinado_coordenador ON
supervisor.id = subordinado_coordenador.subordinado_id
LEFT JOIN prow_usuario supervisor_coordenador ON
supervisor_coordenador.id = subordinado_coordenador.superior_id
LEFT JOIN prow_usuario subordinado subordinado_gerente ON
supervisor_coordenador.id = subordinado_gerente.subordinado_id
LEFT JOIN prow_usuario supervisor_gerente ON supervisor_gerente.id =
subordinado_gerente.superior_id
LEFT JOIN prow_usuario subordinado subordinado_diretor ON
supervisor_gerente.id = subordinado_diretor.subordinado_id
LEFT JOIN prow_usuario supervisor_diretor ON supervisor_diretor.id =
subordinado_diretor.superior_id
LEFT JOIN prow_usuario_perfil perfil ON prow_usuario.perfil_id =
perfil.id
LEFT JOIN prow_regiao ON prow_regiao.empresa_id = prow_empresa.id
WHERE 1 = 1

```

```

AND perfil IN ('PRO',
'VEN')
AND (prow_usuario.inativo = 'false' AND prow_usuario.bloqueado =
'false')
GROUP BY supervisor.codigo, supervisor.nome, prow_usuario.codigo,
prow_usuario.nome, prow_usuario.id, pdvusr.usuario_id, perfil,
prow_empresa.nome, prow_empresa.id
ORDER BY prow_empresa.nome, supervisor.codigo, supervisor.nome,
prow_usuario.codigo, prow_usuario.nome

```

- Select relatório - Visitas e Justificativas.

```

select
    supervisor.codigo as codigoSupervisor,
    supervisor.nome as nomeSupervisor,
    promotor.codigo_codigoPromotor,
    promotor.nome nomePromotor,
    coalesce(visita.tipo_justificativa, 'VISITA') finalidade,
    pdv.codigo_codigoPDV,
    pdv.nome razaoPDV,
    pdv.fantasia fantasiaPDV,
    visita.ts_entrada dataVisita,
    checkin.TS_CAPTURA + interval '1h' * checkin.timezone checkin,
    checkout.TS_CAPTURA + interval '1h' * checkout.timezone
checkout,
    motivo.descricao motivo,
    visita.observacao,
    case when promotor.info_foto = 'false' then
foto_visita.foto_original_id else foto_visita.foto_marca_id end
id_foto,
    visita.id visitaId,
    (select case when it.ponto_venda_id is not null then 'S' else
'N' end
        from prow_agenda ag join prow_item_agenda it on ag.id =
it.agenda_id
        where ag.usuario_id = visita.usuario_id and
it.ponto_venda_id = visita.ponto_venda_id and ag.data BETWEEN
:dataInicio AND :dataFim limit 1) visitaAgenda
from prow_visita visita
join prow_ponto_venda pdv
    on visita.ponto_venda_id = pdv.id
inner join PROW_LOCALIZACAO checkin
    on visita.LOCALIZACAO_CHECKIN_ID = checkin.id
inner join PROW_LOCALIZACAO checkout
    on visita.LOCALIZACAO_CHECKOUT_ID = checkout.id
inner join (select valor configuracao_usuario(prow_usuario.id,
'CONFIG_EXIBIR_INFORMACAO_FOTO') info_foto, prow_usuario.* from
prow_usuario) promotor
    on visita.usuario_id = promotor.id
join prow_usuario_perfil perfil
    on promotor.perfil_id = perfil.id
left join PROW_USUARIO_SUBORDINADO
    on PROW_USUARIO_SUBORDINADO.subordinado_id = promotor.id

left join PROW_USUARIO supervisor
    on PROW_USUARIO_SUBORDINADO.superior_id = supervisor.id
left join prow_motivo_visita motivo
    on visita.motivo_visita_id = motivo.id
left join PROW_FOTO_VISITA foto_visita
    on visita.id = foto_visita.visita_id
where ((1 = :visitaAvulsa and visita.tipo_justificativa =
'VISITA_AVULSA') or (1 = :naoVisita and visita.tipo_justificativa =
'NAO_VISITA')
    or (1 = :checkout and visita.tipo_justificativa = 'CHECKOUT') or
(1 = :visita and visita.tipo_justificativa is null) or (1 = :foraAgenda
and visita.tipo_justificativa = 'FORA_AGENDA'))
and visita.ts_entrada BETWEEN :dataInicio AND :dataFim
AND perfil in ('PRO', 'VEN')
order by nomepromotor, datavisita, finalidade

```

[Voltar](#)

Artigo sugerido pelo analista Dannel Leal.



Não encontrou a solução
que procurava?

[Sugira um artigo](#)



Ainda com dúvida?

[Abra um ticket](#)