================================================

MÉTRICAS E RELATÓRIOS - OFICINA MECÂNICA-- ================================================

================================================

-- 1. MÉTRICAS FINANCEIRAS-- ================================================

-- Faturamento total por período

SELECT

DATE\_FORMAT(dataEmissao, '%Y-%m') AS mes\_ano,

COUNT(\*) AS total\_os,

SUM(valorTotal) AS faturamento\_total,

AVG(valorTotal) AS ticket\_medio

FROM Ordem\_Servico

WHERE status = 'CONCLUIDA'

AND dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 12 MONTH)

GROUP BY DATE\_FORMAT(dataEmissao, '%Y-%m')

ORDER BY mes\_ano DESC;

-- Faturamento por forma de pagamento

SELECT

fp.tipoPagamento,

COUNT(p.idPagamento) AS quantidade\_transacoes,

SUM(p.valor) AS valor\_total,

AVG(p.valor) AS valor\_medio,

ROUND((SUM(p.valor) \* 100.0 / (SELECT SUM(valor) FROM Pagamento)), 2) AS percentual\_faturamento

FROM Pagamento p

JOIN Forma\_Pagamento fp ON p.FormaPagamento\_id = fp.idFormaPagamento

WHERE p.statusPagamento = 'APROVADO'

AND p.dataPagamento >= DATE\_SUB(CURRENT\_DATE, INTERVAL 30 DAY)

GROUP BY fp.tipoPagamento

ORDER BY valor\_total DESC;

-- Top 10 serviços mais lucrativos

SELECT

s.descricao AS servico,

COUNT(os.OS\_id) AS quantidade\_vendida,

SUM(os.valorServico \* os.quantidade) AS receita\_total, AVG(os.valorServico) AS preco\_medio

FROM OS\_Servico os

JOIN Servico s ON os.servico\_id = s.idServicoJOIN Ordem\_Servico o

ON os.OS\_id = o.idOS

WHERE o.status = 'CONCLUIDA' AND o.dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 90 DAY)

GROUP BY s.idServico, s.descricaoORDER BY receita\_total DESC

LIMIT 10;

-- ================================================--

2. MÉTRICAS OPERACIONAIS-- ================================================

-- Status das Ordens de Serviço (atual)

SELECT

status,

COUNT(\*) AS quantidade,

ROUND((COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM Ordem\_Servico)), 2) AS percentual,

SUM(valorTotal) AS valor\_total\_status

FROM Ordem\_Servico

GROUP BY status

ORDER BY quantidade DESC;

-- Tempo médio de conclusão de OS por equipe

SELECT

e.nomeEquipe,

COUNT(os.idOS) AS total\_os\_concluidas,

AVG(TIMESTAMPDIFF(HOUR, os.dataEmissao, os.dataConclusao)) AS tempo\_medio\_horas,

MIN(TIMESTAMPDIFF(HOUR, os.dataEmissao, os.dataConclusao)) AS tempo\_minimo\_horas,

MAX(TIMESTAMPDIFF(HOUR, os.dataEmissao, os.dataConclusao)) AS tempo\_maximo\_horas

FROM Ordem\_Servico os

JOIN Equipe e ON os.equipe\_id = e.idEquipe

WHERE os.status = 'CONCLUIDA'

AND os.dataConclusao IS NOT NULL

AND os.dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 90 DAY)

GROUP BY e.idEquipe, e.nomeEquipe

ORDER BY tempo\_medio\_horas ASC;

-- Produtividade por mecânico (baseado nas equipes)

SELECT

m.nome AS mecanico,

e.nomeEquipe,

COUNT(os.idOS) AS os\_participou,

SUM(os.valorTotal) AS valor\_total\_gerado,

AVG(os.valorTotal) AS ticket\_medio

FROM Mecanico m

JOIN Equipe\_Mecanico em ON m.idMecanico = em.mecanico\_idJOIN Equipe e ON em.equipe\_id = e.idEquipe

JOIN Ordem\_Servico os ON e.idEquipe = os.equipe\_id

WHERE os.status = 'CONCLUIDA'

AND os.dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 30 DAY)

GROUP BY m.idMecanico, m.nome, e.nomeEquipe

ORDER BY valor\_total\_gerado DESC;--

================================================

-- 3. MÉTRICAS DE CLIENTE-- ================================================

Top 10 clientes por faturamento

SELECT

c.nome AS cliente,

c.CPF,

COUNT(DISTINCT os.idOS) AS total\_os,

SUM(os.valorTotal) AS faturamento\_total,

AVG(os.valorTotal) AS ticket\_medio,

MAX(os.dataEmissao) AS ultima\_visita

FROM Cliente c

JOIN Veiculo v ON c.idCliente = v.cliente\_idJOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo

WHERE os.status = 'CONCLUIDA'

GROUP BY c.idCliente, c.nome, c.CPF

ORDER BY faturamento\_total DESC

LIMIT 10;

-- Frequência de retorno dos clientes

SELECT

'Clientes Novos (1 OS)' AS categoria,

COUNT(\*) AS quantidade

FROM ( SELECT c.idCliente

FROM Cliente c

JOIN Veiculo v ON c.idCliente = v.cliente\_id

JOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo

WHERE os.status = 'CONCLUIDA'

GROUP BY c.idCliente

HAVING COUNT(os.idOS) = 1) AS novos

UNION ALL

SELECT

'Clientes Recorrentes (2-5 OS)' AS categoria,

COUNT(\*) AS quantidade

FROM ( SELECT c.idCliente

FROM Cliente c

JOIN Veiculo v ON c.idCliente = v.cliente\_id

JOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo

WHERE os.status = 'CONCLUIDA'

GROUP BY c.idCliente

HAVING COUNT(os.idOS) BETWEEN 2 AND 5) AS recorrentes

UNION ALL

SELECT 'Clientes Fiéis (6+ OS)' AS categoria,

COUNT(\*) AS quantidade

FROM ( SELECT c.idCliente

FROM Cliente c

JOIN Veiculo v ON c.idCliente = v.cliente\_id

JOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo

WHERE os.status = 'CONCLUIDA'

GROUP BY c.idCliente

HAVING COUNT(os.idOS) >= 6) AS fieis;

-- ================================================--

4. MÉTRICAS DE ESTOQUE E PEÇAS-- ================================================

-- Peças mais utilizadas

SELECT

p.nome AS peca,

SUM(op.quantidade) AS quantidade\_total\_utilizada,

SUM(op.valorPeca \* op.quantidade) AS receita\_pecas,

AVG(op.valorPeca) AS preco\_medio\_venda,

p.estoque AS estoque\_atual

FROM OS\_Peca op

JOIN Peca p ON op.peca\_id = p.idPeca

JOIN Ordem\_Servico os ON op.OS\_id = os.idOS

WHERE os.status = 'CONCLUIDA'

AND os.dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 90 DAY)

GROUP BY p.idPeca, p.nome, p.estoque

ORDER BY quantidade\_total\_utilizada DESC

LIMIT 10;

-- Controle de estoque - peças em baixa

SELECT

nome AS peca,

estoque AS estoque\_atual,

valorUnitario AS custo\_unitario,

(estoque \* valorUnitario) AS valor\_estoque\_atual,

CASE

WHEN estoque = 0 THEN 'SEM ESTOQUE'

WHEN estoque <= 5 THEN 'CRÍTICO'

WHEN estoque <= 10 THEN 'BAIXO'

ELSE 'OK'

END AS status\_estoque

FROM Peca

WHERE estoque <= 10

ORDER BY estoque ASC, valorUnitario DESC;

-- ================================================

-- 5. MÉTRICAS DE VEÍCULOS-- ================================================--

Marcas/modelos mais atendidos

SELECT

v.marca,

v.modelo,

COUNT(DISTINCT v.placa) AS total\_veiculos\_cadastrados,

COUNT(os.idOS) AS total\_atendimentos,

SUM(os.valorTotal) AS faturamento\_marca\_modelo,

AVG(os.valorTotal) AS ticket\_medio

FROM Veiculo v

LEFT JOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo AND os.status = 'CONCLUIDA'

GROUP BY v.marca, v.modelo

HAVING COUNT(os.idOS) > 0

ORDER BY total\_atendimentos DESC

LIMIT 15;

-- Análise por ano do veículo

SELECT

v.ano,

COUNT(DISTINCT v.placa) AS total\_veiculos,

COUNT(os.idOS) AS total\_atendimentos,

SUM(os.valorTotal) AS faturamento\_total,

AVG(os.valorTotal) AS ticket\_medio,

(YEAR(CURRENT\_DATE) - v.ano) AS idade\_media

FROM Veiculo v

LEFT JOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo AND os.status = 'CONCLUIDA'

WHERE os.idOS IS NOT NULL

GROUP BY v.ano

ORDER BY v.ano DESC;

-- ================================================--

6. DASHBOARD EXECUTIVO (RESUMO)-- ================================================

-- KPIs principais do mês atual

SELECT

'Faturamento Mês Atual' AS indicador,

CONCAT('R$ ', FORMAT(SUM(valorTotal), 2)) AS valor

FROM Ordem\_Servico

WHERE status = 'CONCLUIDA'

AND MONTH(dataEmissao) = MONTH(CURRENT\_DATE)

AND YEAR(dataEmissao) = YEAR(CURRENT\_DATE)

UNION ALL

SELECT

'OS Concluídas Mês Atual' AS indicador,

COUNT(\*) AS valor

FROM Ordem\_Servico

WHERE status = 'CONCLUIDA'

AND MONTH(dataEmissao) = MONTH(CURRENT\_DATE)

AND YEAR(dataEmissao) = YEAR(CURRENT\_DATE)

UNION ALL

SELECT

'Ticket Médio Mês Atual' AS indicador,

CONCAT('R$ ', FORMAT(AVG(valorTotal), 2)) AS valor

FROM Ordem\_Servico

WHERE status = 'CONCLUIDA'

AND MONTH(dataEmissao) = MONTH(CURRENT\_DATE)

AND YEAR(dataEmissao) = YEAR(CURRENT\_DATE)

UNION ALL

SELECT

'OS Pendentes' AS indicador,

COUNT(\*) AS valor

FROM Ordem\_Servico

WHERE status IN ('ABERTA', 'EM\_ANDAMENTO')

UNION ALL

SELECT 'Clientes Ativos (últimos 30 dias)' AS indicador,

COUNT(DISTINCT c.idCliente) AS valor

FROM Cliente c

JOIN Veiculo v ON c.idCliente = v.cliente\_id

JOIN Ordem\_Servico os ON v.placa = os.placa\_veiculo

WHERE os.dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 30 DAY);

-- ================================================--

7. ANÁLISE COMPARATIVA (MÊS ATUAL VS ANTERIOR)-- ================================================

SELECT

'Faturamento' AS metrica,

SUM(CASE

WHEN MONTH(dataEmissao) = MONTH(CURRENT\_DATE)

AND YEAR(dataEmissao) = YEAR(CURRENT\_DATE)

THEN valorTotal ELSE 0 END) AS mes\_atual,

SUM(CASE

WHEN MONTH(dataEmissao) = MONTH(DATE\_SUB(CURRENT\_DATE, INTERVAL 1 MONTH))

AND YEAR(dataEmissao) = YEAR(DATE\_SUB(CURRENT\_DATE, INTERVAL 1 MONTH))

THEN valorTotal ELSE 0 END) AS mes\_anterior,

ROUND(((SUM(CASE

WHEN MONTH(dataEmissao) = MONTH(CURRENT\_DATE)

AND YEAR(dataEmissao) = YEAR(CURRENT\_DATE)

THEN valorTotal ELSE 0 END) -

SUM(CASE

WHEN MONTH(dataEmissao) = MONTH(DATE\_SUB(CURRENT\_DATE, INTERVAL 1 MONTH))

AND YEAR(dataEmissao) = YEAR(DATE\_SUB(CURRENT\_DATE, INTERVAL 1 MONTH))

THEN valorTotal ELSE 0 END)) /

NULLIF(SUM(CASE

WHEN MONTH(dataEmissao) = MONTH(DATE\_SUB(CURRENT\_DATE, INTERVAL 1 MONTH))

AND YEAR(dataEmissao) = YEAR(DATE\_SUB(CURRENT\_DATE, INTERVAL 1 MONTH))

THEN valorTotal ELSE 0 END), 0) \* 100), 2) AS variacao\_percentual

FROM Ordem\_Servico

WHERE status = 'CONCLUIDA'

AND dataEmissao >= DATE\_SUB(CURRENT\_DATE, INTERVAL 2 MONTH);