

Rolling Revenue Report for Previous Years

Last Modified on 17/02/2026 2:16 pm EST

/*

This query provides monthly operator-level summary of bookings, WH Out, invoiced revenue, averages, closure ratios, lost revenue, and prior-year revenue for trend analysis over a rolling multi-year period.

*/

/*

Key Features & Metrics:

1. Booking Totals (Created & WH Out)

- [Booking Total Created] – total revenue (net of tax) from bookings based on the entry date.
- [Booking Total WH Out] – total revenue (net of tax) based on the WH date (ddate).

2. Invoiced Metrics

- [Invoice Total] and [Qty Invoiced] – revenue and count for fully invoiced bookings.

3. Averages

- [Average Booking Created] – average revenue per booking for created bookings.
- [Average WH Out] – average revenue per WH Out booking.
- [Average Invoice] – average revenue per invoiced booking.

4. Ratios & Performance Indicators

- [Ratio Closed Quantity] – % of WH Out bookings that were invoiced.
- [Ratio Closed Revenue] – % of WH Out revenue that was invoiced.
- [Revenue Lost] – difference between WH Out revenue and invoiced revenue (unclosed bookings).

5. Historical Comparison

- [Revenue Last Year] – invoiced revenue for the same period in the previous year, aligned to allow trend analysis.

6. Rolling Years & Filters

- Rolling window controlled by `@RollingYears` parameter.
- Revenue calculations exclude certain booking types as defined.

Output: One row per operator per month, including all totals, averages, ratios, lost revenue, and previous year revenue, ready for reporting or KPI dashboards.

*/

Running the Report

/* Complete step 2 before trying to use this report.

Add the code below to [EXCEL Query Builder](#)

Change the values for `@Year` and `@RollingYears` as needed */

```
EXEC dbo.usp_OperatorBookingReport
```

```
    @Year = 2025,
```

```
    @RollingYears = 4;
```

BEFORE Running the Report

/* Paste the code below into SQL Server Management Studio and execute */

```
CREATE PROCEDURE dbo.usp_OperatorBookingReport
```

```
    @Year INT,          -- Ending year for rolling window
```

```
    @RollingYears INT  -- Number of years to go back
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    -- Calculate date ranges
```

```
    DECLARE @StartDate DATE = DATEADD(YEAR, -@RollingYears, DATEFROMPARTS(@Year,1,1));
```

```
    DECLARE @EndDate DATE = DATEFROMPARTS(@Year,12,31);
```

```
    -- Previous Year Window
```

```
    DECLARE @LYStartDate DATE = DATEADD(YEAR, -1, @StartDate);
```

```
DECLARE @LYEndDate DATE = DATEADD(YEAR, -1, @EndDate);
```

```
;WITH Base AS
```

```
(
```

```
SELECT
```

```
    b.booking_no,
```

```
    b.ddate,
```

```
    b.entrydate,
```

```
    b.booking_type_v32,
```

```
    b.BookingProgressStatus,
```

```
    b.invoiced,
```

```
    o.firstname AS Operator,
```

```
    NetAmount = b.price_quoted - ISNULL(b.tax1,0) - ISNULL(b.tax2,0)
```

```
FROM dbo.tblbookings b
```

```
LEFT JOIN dbo.tbloperators o
```

```
    ON b.operatorsid = o.id
```

```
),
```

```
-- CREATED (entrydate driven)
```

```
CreatedStream AS
```

```
(
```

```
SELECT
```

```
    Operator,
```

```
    YEAR(ddate) AS [Year],
```

```
    MONTH(ddate) AS MonthNumber,
```

```
    DATENAME(MONTH, ddate) AS [Month],
```

```
    NetAmount AS CreatedTotal,
```

```
    1 AS CreatedQty,
```

0 AS WHOutTotal,

0 AS WHOutQty,

0 AS InvoiceTotal,

0 AS InvoiceQty,

0 AS RevenueLastYear

FROM Base

WHERE entrydate BETWEEN @StartDate AND @EndDate

AND booking_type_v32 NOT IN (5,11,13)

),

-- WH OUT + INVOICE (ddate driven)

WHStream AS

(

SELECT

Operator,

YEAR(ddate) AS [Year],

MONTH(ddate) AS MonthNumber,

DATENAME(MONTH, ddate) AS [Month],

0 AS CreatedTotal,

0 AS CreatedQty,

CASE WHEN booking_type_v32 NOT IN (5,11,13) THEN NetAmount ELSE 0 END AS
WHOutTotal,

CASE WHEN booking_type_v32 NOT IN (5,11,13) THEN 1 ELSE 0 END AS WHOutQty,

CASE WHEN BookingProgressStatus = 3

AND invoiced = 'Y'

AND booking_type_v32 NOT IN (1,5,11,13)

THEN NetAmount ELSE 0 END AS InvoiceTotal,

CASE WHEN BookingProgressStatus = 3

```

        AND invoiced = 'Y'

        AND booking_type_v32 NOT IN (1,5,11,13)

    THEN 1 ELSE 0 END AS InvoiceQty,

    0 AS RevenueLastYear

FROM Base

WHERE ddate BETWEEN @StartDate AND @EndDate

),

-- LAST YEAR REVENUE

LastYearStream AS

(

    SELECT

        Operator,

        YEAR(DATEADD(YEAR,1,ddate)) AS [Year], -- shift forward for alignment

        MONTH(ddate) AS MonthNumber,

        DATENAME(MONTH, ddate) AS [Month],

        0 AS CreatedTotal,

        0 AS CreatedQty,

        0 AS WHOutTotal,

        0 AS WHOutQty,

        0 AS InvoiceTotal,

        0 AS InvoiceQty,

        NetAmount AS RevenueLastYear

    FROM Base

    WHERE ddate BETWEEN @LYStartDate AND @LYEndDate

        AND BookingProgressStatus = 3

        AND invoiced = 'Y'

        AND booking_type_v32 NOT IN (1,5,11,13)

```

)

-- FINAL AGGREGATION

SELECT

Operator,

[Year],

[Month],

-- CREATED

ISNULL(SUM(CreatedTotal),0) AS [Booking Total-C],

ISNULL(SUM(CreatedQty),0) AS [Qty Booked-C],

ISNULL(CAST(SUM(CreatedTotal)/NULLIF(SUM(CreatedQty),0) AS DECIMAL(10,2)),0) AS
[Average Booking-C],

-- WH OUT

ISNULL(SUM(WHOutTotal),0) AS [Booking Total-O],

ISNULL(SUM(WHOutQty),0) AS [Qty Booked-O],

ISNULL(CAST(SUM(WHOutTotal)/NULLIF(SUM(WHOutQty),0) AS DECIMAL(10,2)),0) AS
[Average Booking-O],

-- INVOICE

ISNULL(SUM(InvoiceTotal),0) AS [Invoice Total],

ISNULL(SUM(InvoiceQty),0) AS [Qty Invoiced],

ISNULL(CAST(SUM(InvoiceTotal)/NULLIF(SUM(InvoiceQty),0) AS DECIMAL(10,2)),0) AS
[Average Invoice],

-- RATIOS

ISNULL(CAST(SUM(InvoiceQty)/NULLIF(SUM(WHOutQty),0) AS DECIMAL(10,4)),0) AS [Ratio
Closed Quantity],

```
ISNULL(CAST(SUM(InvoiceTotal)/NULLIF(SUM(WHOutTotal),0) AS DECIMAL(10,4)),0) AS [Ratio  
Closed Revenue],
```

```
-- REVENUE LOST
```

```
ISNULL(SUM(WHOutTotal),0) - ISNULL(SUM(InvoiceTotal),0) AS [Revenue Lost],
```

```
-- REVENUE LAST YEAR
```

```
ISNULL(SUM(RevenueLastYear),0) AS [Revenue Last Year]
```

```
FROM
```

```
(
```

```
SELECT * FROM CreatedStream
```

```
UNION ALL
```

```
SELECT * FROM WHStream
```

```
UNION ALL
```

```
SELECT * FROM LastYearStream
```

```
) Combined
```

```
GROUP BY
```

```
Operator,
```

```
[Year],
```

```
MonthNumber,
```

```
[Month]
```

```
ORDER BY
```

```
[Year],
```

```
MonthNumber,
```

```
Operator;
```

END

GO