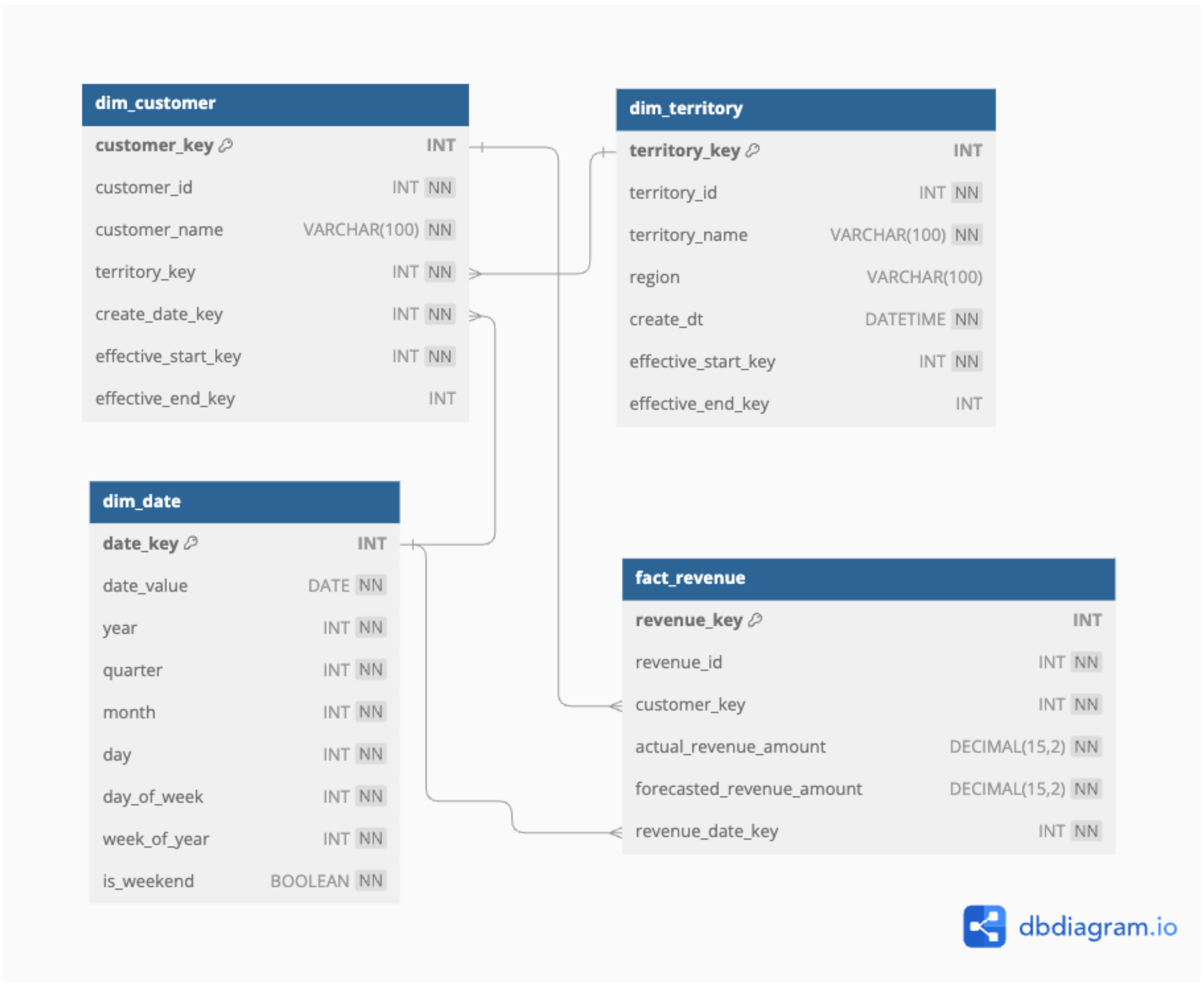
Entity-Level Dimensional Data Model for Tracking Customer Revenue by Sales Territory and Time

Tables and Fields:

1. **dim_customer** (Dimension Table for Customers)
   - **customer_key** (Primary Key): Unique identifier for each customer.
   - **customer_id**: Identifier for the customer, probably from an external source.
   - **customer_name**: Name of the customer, stored as a VARCHAR of length 100.
   - **territory_key**: Foreign key reference to the `dim_territory` table.
   - **create_date_key**: Foreign key reference to the `dim_date` table, marking the date of customer creation.
   - **effective_start_key** and **effective_end_key**: Foreign key references to the `dim_date` table, likely marking the start and end dates of a customer's effectiveness.
2. **dim_territory** (Dimension Table for Territories)
   - **territory_key** (Primary Key): Unique identifier for each territory.
   - **territory_id**: Identifier for the territory, likely from an external system.
   - **territory_name**: Name of the territory, stored as a VARCHAR of length 100.
   - **region**: Region of the territory, stored as a VARCHAR of length 100.
   - **create_dt**: Creation date for the territory record, stored as DATETIME.
   - **effective_start_key** and **effective_end_key**: Foreign key references to `dim_date`, marking the effective start and end dates for the territory.
3. **dim_date** (Date Dimension Table)
   - **date_key** (Primary Key): Unique identifier for each date.
   - **date_value**: Actual date, stored in DATE format.
   - **year, quarter, month, day, day_of_week, week_of_year**: Breakdown of date components for easier querying and reporting.
   - **is_weekend**: Boolean flag indicating if the date falls on a weekend.
4. **fact_revenue** (Fact Table for Revenue)
   - **revenue_key** (Primary Key): Unique identifier for each revenue record.
   - **revenue_id**: Identifier for revenue transactions, possibly from an external system.
   - **customer_key**: Foreign key reference to `dim_customer`, linking revenue to specific customers.
   - **amount**: Revenue amount, stored as DECIMAL with a precision of 15 and scale of 2.
   - **revenue_date_key**: Foreign key reference to `dim_date`, representing the date of revenue generation.

Relationships:

- **dim_customer** is linked to **dim_territory** through the `territory_key` field, which establishes the relationship between customers and their respective territories.
- **dim_customer** also has multiple references to **dim_date** via `create_date_key`, `effective_start_key`, and `effective_end_key`, which capture various date-based information for customers.
- **dim_territory** is similarly linked to **dim_date** using `effective_start_key` and `effective_end_key`, denoting the valid time range of each territory.
- **fact_revenue** is the central fact table, connecting to **dim_customer** via `customer_key` and **dim_date** via `revenue_date_key`, allowing the analysis of revenue by customer and date.



dim_customer		
customer_key	**INT**	
customer_id	INT	NN
customer_name	VARCHAR(100)	NN
territory_key	INT	NN
create_date_key	INT	NN
effective_start_key	INT	NN
effective_end_key	INT	

dim_territory		
territory_key	**INT**	
territory_id	INT	NN
territory_name	VARCHAR(100)	NN
region	VARCHAR(100)	
create_dt	DATETIME	NN
effective_start_key	INT	NN
effective_end_key	INT	

dim_date		
date_key	**INT**	
date_value	DATE	NN
year	INT	NN
quarter	INT	NN
month	INT	NN
day	INT	NN
day_of_week	INT	NN
week_of_year	INT	NN
is_weekend	BOOLEAN	NN

fact_revenue		
revenue_key	**INT**	
revenue_id	INT	NN
customer_key	INT	NN
actual_revenue_amount	DECIMAL(15,2)	NN
forecasted_revenue_amount	DECIMAL(15,2)	NN
revenue_date_key	INT	NN

Actual Revenue by Sales Territory

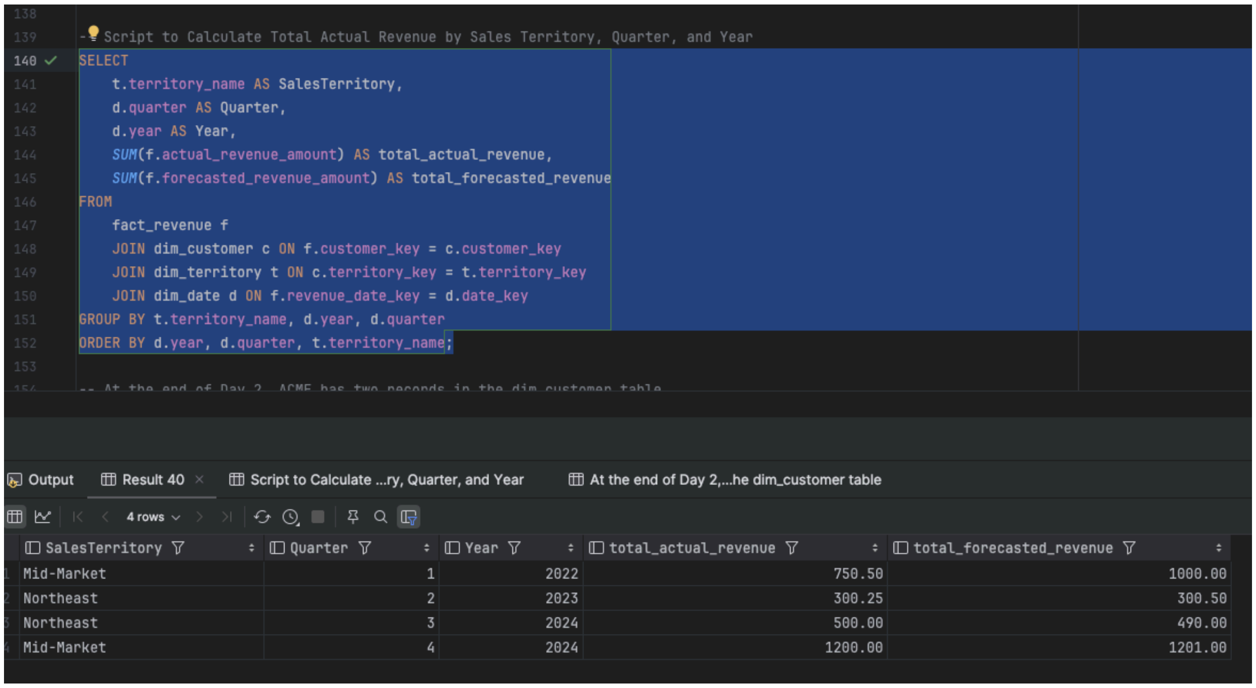
Bar Chart

- **X-Axis**: **Sales Territory** (e.g., Northeast, Mid-Market, Southwest)
- **Y-Axis**: **Total Actual Revenue** (aggregate revenue value)
- **Bars**: Each bar represents the **total revenue** for a specific **Sales Territory**
- **Color Coding (Optional)**: Different colors for each quarter to show trends within each territory over time.

Visualization Explanation

- **Sales Territory**: Indicates the name of each sales region.
- **Quarter** and **Year**: Shows the specific time period for the revenue figures.
- **Total Actual Revenue**: Displays the sum of actual revenue for each territory per quarter, helping stakeholders compare performance across territories and time periods.

Script to Calculate Total Actual Revenue by Sales Territory, Quarter, and Year



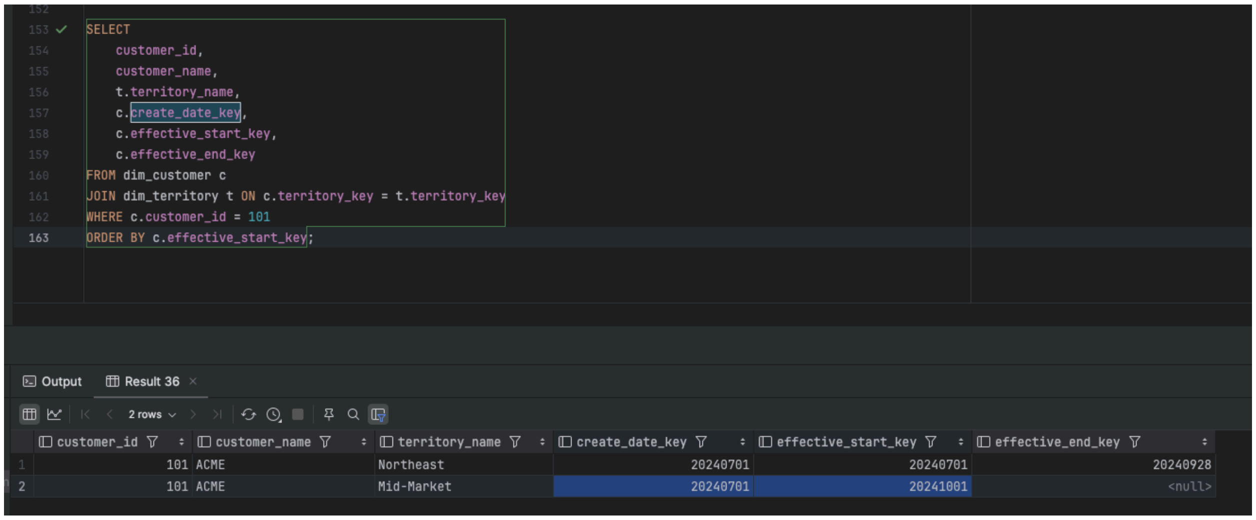
```sql
-- Script to Calculate Total Actual Revenue by Sales Territory, Quarter, and Year
SELECT
    t.territory_name AS SalesTerritory,
    d.quarter AS Quarter,
    d.year AS Year,
    SUM(f.actual_revenue_amount) AS total_actual_revenue,
    SUM(f.forecasted_revenue_amount) AS total_forecasted_revenue
FROM
    fact_revenue f
    JOIN dim_customer c ON f.customer_key = c.customer_key
    JOIN dim_territory t ON c.territory_key = t.territory_key
    JOIN dim_date d ON f.revenue_date_key = d.date_key
GROUP BY t.territory_name, d.year, d.quarter
ORDER BY d.year, d.quarter, t.territory_name;

-- At the end of Day 2, ACME has two records in the dim_customer table
```

Output
Result 40
Script to Calculate ...ry, Quarter, and Year
At the end of Day 2,...he dim_customer table
4 rows

	SalesTerritory	Quarter	Year	total_actual_revenue	total_forecasted_revenue
1	Mid-Market	1	2022	750.50	1000.00
2	Northeast	2	2023	300.25	300.50
3	Northeast	3	2024	500.00	490.00
4	Mid-Market	4	2024	1200.00	1201.00

At the end of Day 2, ACME has two records in the dim_customer table:

1. One record shows ACME in the **Northeast** territory, effective from 2024-07-01 to 2024-10-01.
2. The second record shows ACME realigned to the **Mid-Market** territory, starting from 2024-10-01 with no effective_end_key, indicating it's the current alignment.



```sql
SELECT
    customer_id,
    customer_name,
    t.territory_name,
    c.create_date_key,
    c.effective_start_key,
    c.effective_end_key
FROM dim_customer c
JOIN dim_territory t ON c.territory_key = t.territory_key
WHERE c.customer_id = 101
ORDER BY c.effective_start_key;
```

	customer_id	customer_name	territory_name	create_date_key	effective_start_key	effective_end_key
1	101	ACME	Northeast	20240701	20240701	20240928
2	101	ACME	Mid-Market	20240701	20241001	<null>

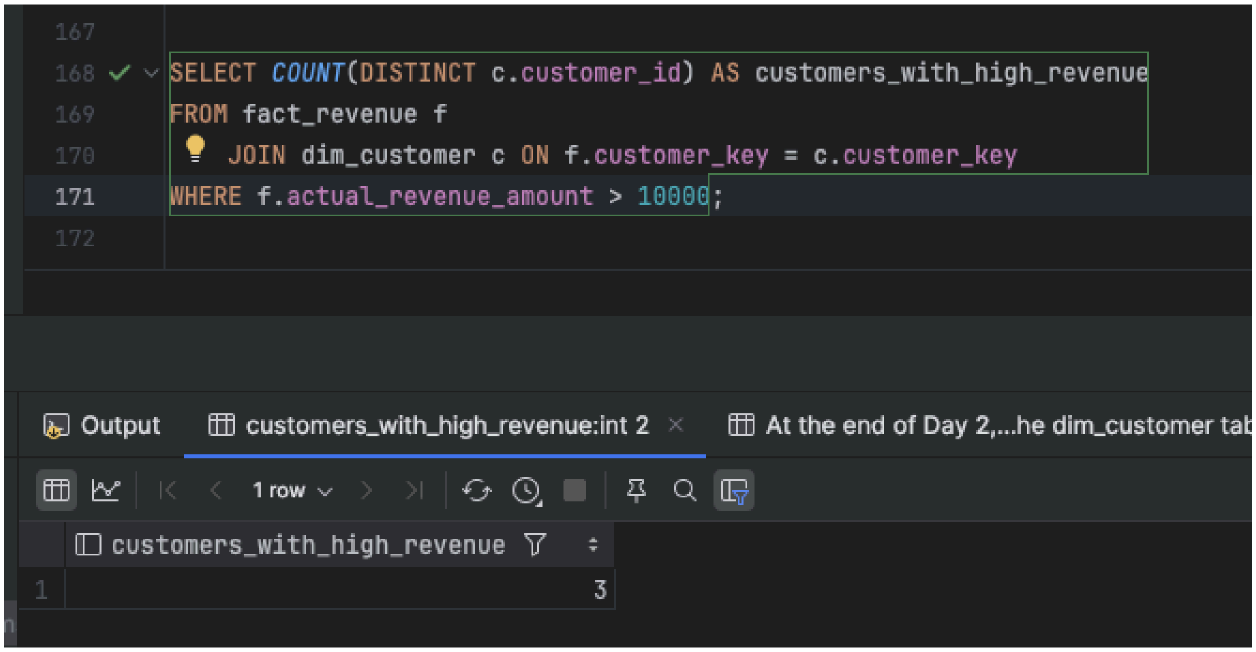
Join Forecasted Revenue with Actuals

The dim_customer table should use customer_id instead of customer_key as the unique identifier for joins. The join condition should be written as:

ar.customer_id = c.customer_id AND ar.revenue_date_key BETWEEN c.effective_start_key AND COALESCE(c.effective_end_key, 30000101)
This allows a view to be created that can be easily joined.

Alternatively, in this case, we can separate it into two tables: **`revenue`** and **`forecasted`** `revenue`.

Count of Distinct Customers with Revenue > $10K:

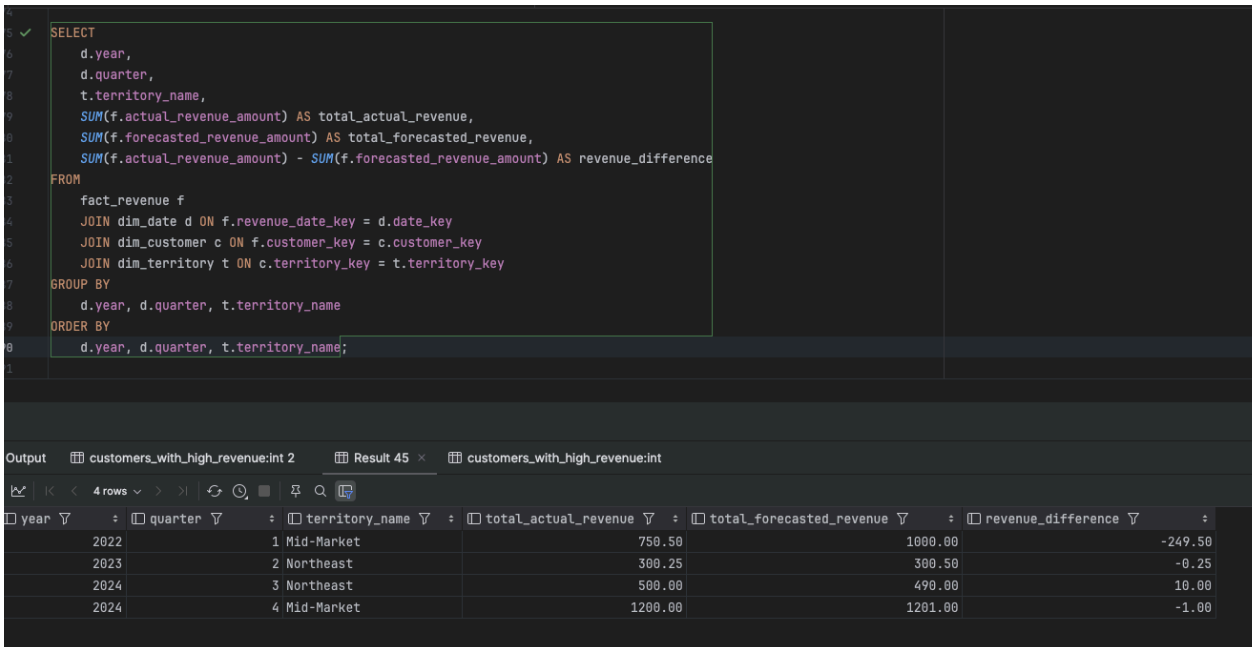


```sql
SELECT COUNT(DISTINCT c.customer_id) AS customers_with_high_revenue
FROM fact_revenue f
 JOIN dim_customer c ON f.customer_key = c.customer_key
WHERE f.actual_revenue_amount > 10000;
```

Output
customers_with_high_revenue:int 2
At the end of Day 2,...he dim_customer tab
1 row

```
  customers_with_high_revenue
1                           3
```

Sum of Actual Revenue, Forecasted Revenue, and the Difference for Each by Quarter & Territory

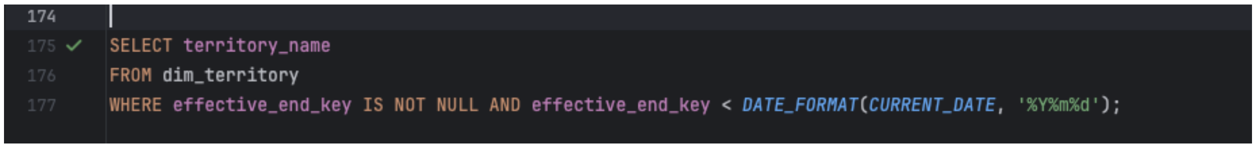


```sql
SELECT
    d.year,
    d.quarter,
    t.territory_name,
    SUM(f.actual_revenue_amount) AS total_actual_revenue,
    SUM(f.forecasted_revenue_amount) AS total_forecasted_revenue,
    SUM(f.actual_revenue_amount) - SUM(f.forecasted_revenue_amount) AS revenue_difference
FROM
    fact_revenue f
    JOIN dim_date d ON f.revenue_date_key = d.date_key
    JOIN dim_customer c ON f.customer_key = c.customer_key
    JOIN dim_territory t ON c.territory_key = t.territory_key
GROUP BY
    d.year, d.quarter, t.territory_name
ORDER BY
    d.year, d.quarter, t.territory_name;
```

Output
customers_with_high_revenue:int 2
Result 45
customers_with_high_revenue:int
4 rows

```
year  quarter  territory_name  total_actual_revenue  total_forecasted_revenue  revenue_difference
2022        1  Mid-Market                    750.50                   1000.00             -249.50
2023        2  Northeast                     300.25                    300.50               -0.25
2024        3  Northeast                     500.00                    490.00               10.00
2024        4  Mid-Market                   1200.00                   1201.00               -1.00
```

Return Territories That No Longer Exist as of Today (But Did Exist Previously)



```sql

SELECT territory_name
FROM dim_territory
WHERE effective_end_key IS NOT NULL AND effective_end_key < DATE_FORMAT(CURRENT_DATE, '%Y%m%d');
```

This script creates a data model with tables for date, territory, customer, and revenue, including sample data and a procedure to populate dates.

Key queries:

1. Calculate actual vs. forecasted revenue by customer, quarter, and territory.
2. Summarize revenue by territory and year.
3. Count customers with revenue > $10K.
4. Identify inactive territories.

These queries enable revenue analysis, customer performance tracking, and territory management insights.