Step-by-Step Explanation of the written code

1.Import Required Libraries

```
import streamlit as st
import pandas as pd
import pyodbc
import sys
```

- import streamlit as st: Builds the interactive dashboard.
- import pandas as pd: Handles table data.
- import pyodbc: Connects Python to SQL Server.

2. Connect to SQL Server

```
# SQL Server connection
conn = pyodbc.connect(
    "DRIVER={ODBC Driver 17 for SQL Server};"
    "SERVER=DESKTOP-0F9OCUO;"  # Use your instance name if different
    "DATABASE=SR;"            # Change to your actual database name
    "Trusted_Connection=yes;"
)
```

- DRIVER: Tells Python which SQL driver to use.
- SERVER: The location of the SQL Server instance.
- DATABASE: The name of the restored database.
- Trusted_Connection: Uses your Windows login credentials.

3. Load Data from SQL Table

```
# Load data from your table
df = pd.read_sql("SELECT * FROM [Employee Data]", conn)

df["Active"] = df["Active?"] == "Y"
```

- Loads all rows from the `Employee Data` table into a pandas DataFrame called `df`.

4. Create an 'Active' Column

- Converts "Y"/"N" from `Active?` column into `True`/`False` for easier filtering.

5. Add Sidebar Filters

```
# Sidebar filters
st.sidebar.header("Filter Employees")
selected_role = st.sidebar.selectbox("Select Role", ["All"] + sorted(df["Role"].unique()))
selected_location = st.sidebar.selectbox("Select Location", ["All"] + sorted(df["Location"].unique()))
show_inactive = st.sidebar.checkbox("Include Inactive Employees", value=True)
```

- Adds interactive filters for Role, Location, and Active status using Streamlit sidebar.

6. Apply Filters to the Data

```
# Apply filters
filtered_df = df.copy()
if selected_role != "All":
    filtered_df = filtered_df[filtered_df["Role"] == selected_role]
if selected_location != "All":
    filtered_df = filtered_df[filtered_df["Location"] == selected_location]
if not show_inactive:
    filtered_df = filtered_df[filtered_df["Active"]]
```

- Apply filters to the DataFrame `filtered_df` based on user selections.

7. Show Filtered Employee Table

```
# View filtered data
st.title("Employee Dashboard")
st.dataframe(filtered_df[["Name", "Role", "Location", "Current Comp (INR)"]])
```

- Displays the filtered data in a table using Streamlit.

8. Show Average Compensation for Selected Location

```
# Avg compensation
if selected_location != "All":
    avg = filtered_df["Current Comp (INR)"].mean()
    st.metric(label=f"Average Compensation in {selected_location}", value=f"₹{avg:,.0f}")
```

- Displays the average salary in a metric box if a location is selected.

9. Bar Chart: Average Compensation by Location

```
    # Avg compensation
    if selected_location != "All":
        avg = filtered_df["Current Comp (INR)"].mean()
        st.metric(label=f"Average Compensation in {selected_location}", value=f"₹{avg:,.0f}")
```

- Plots a bar chart of average salary grouped by location.

10. Bar Chart: Employees by Experience Range

```
# Experience breakdown
st.subheader("Employees by Experience Range")
st.bar_chart(filtered_df["Years of Experience"].value_counts().sort_index())
```

- Displays how many employees fall in each experience range.

11. Simulate Global Compensation Increment

```
# Variable increment
st.subheader("Simulate Compensation Increment")
increment = st.slider("Select Global % Increment", 0, 100, 10)
filtered_df["Incremented Compensation"] = filtered_df["Current Comp (INR)"] * (1 + increment / 100)
st.dataframe(filtered_df[["Name", "Current Comp (INR)", "Incremented Compensation"]])
st.success(f"Total Updated Compensation: ₹{filtered_df['Incremented Compensation'].sum():,.0f}")
```

- Adds a slider to simulate a global salary raise.
- Creates a new column with updated compensation.

12. Show Total Updated Compensation in a Box

```
# Export CSV
st.download_button("Download CSV", filtered_df.to_csv(index=False), file_name="filtered_employees.csv")
```

- Displays total updated compensation in a green info box.

13. Export Filtered Data to CSV

- Adds a download button to export the filtered and updated data to CSV.