Task 2: Advanced Data Analysis

Step 1: Create the Database

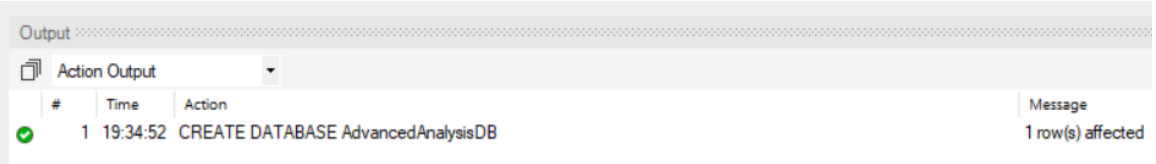
Query:

CREATE DATABASE AdvancedAnalysisDB;

Explanation:

This query creates a new database named 'AdvancedAnalysisDB'.

Output:



```
   #   Time      Action                                    Message
   1   19:34:52  CREATE DATABASE AdvancedAnalysisDB        1 row(s) affected
```

Step 2: Use the Created Database

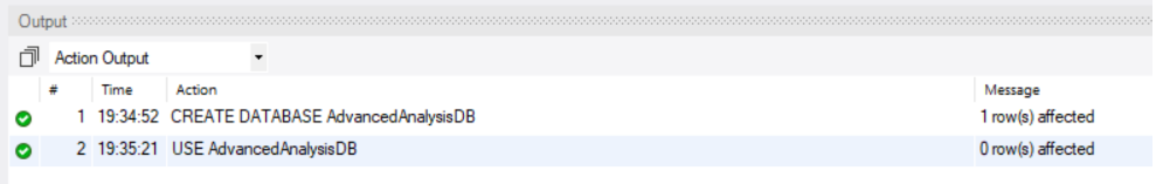
Query:

USE AdvancedAnalysisDB;

Explanation:

Switches the active database context to 'AdvancedAnalysisDB'.

Output:



```
   #   Time      Action                                    Message
   1   19:34:52  CREATE DATABASE AdvancedAnalysisDB        1 row(s) affected
   2   19:35:21  USE AdvancedAnalysisDB                    0 row(s) affected
```

Step 3: Create the Employees Table

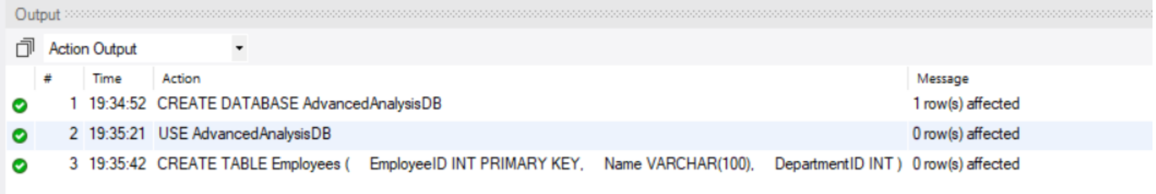
Query:

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    DepartmentID INT
);
```

Explanation:

Creates a table named 'Employees' with columns for Employee ID, Name, and Department ID.

Output:



	#	Time	Action	Message
✓	1	19:34:52	CREATE DATABASE AdvancedAnalysisDB	1 row(s) affected
✓	2	19:35:21	USE AdvancedAnalysisDB	0 row(s) affected
✓	3	19:35:42	CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR(100), DepartmentID INT)	0 row(s) affected

Step 4: Insert Data into Employees Table

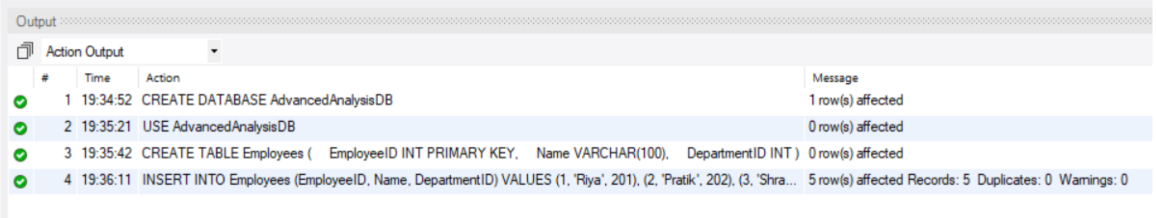
Query:

```
INSERT INTO Employees (EmployeeID, Name, DepartmentID) VALUES
(1, 'Riya', 201),
(2, 'Pratik', 202),
(3, 'Shravani', 203),
(4, 'Kartik', 204),
(5, 'Pratiksha', NULL);
```

Explanation:

Populates the 'Employees' table with sample data.

Output:



	#	Time	Action	Message
✓	1	19:34:52	CREATE DATABASE AdvancedAnalysisDB	1 row(s) affected
✓	2	19:35:21	USE AdvancedAnalysisDB	0 row(s) affected
✓	3	19:35:42	CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR(100), DepartmentID INT)	0 row(s) affected
✓	4	19:36:11	INSERT INTO Employees (EmployeeID, Name, DepartmentID) VALUES (1, 'Riya', 201), (2, 'Pratik', 202), (3, 'Shra...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

Step 5: Create the Departments Table

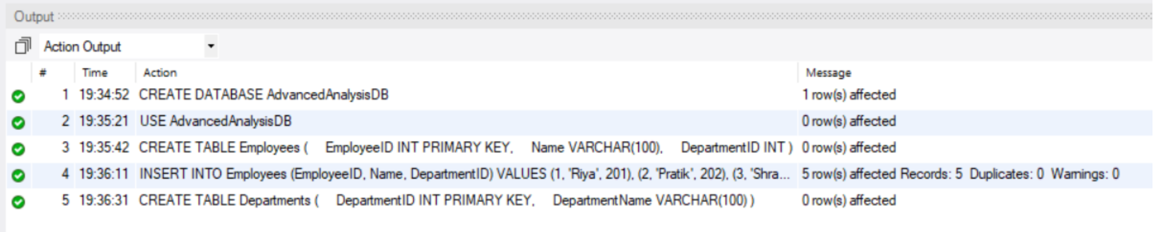
Query:

```
CREATE TABLE Departments (
   DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100));
```

Explanation:

Creates a table named 'Departments' with columns for Department ID and Department Name.

Output:



	#	Time	Action	Message
✓	1	19:34:52	CREATE DATABASE AdvancedAnalysisDB	1 row(s) affected
✓	2	19:35:21	USE AdvancedAnalysisDB	0 row(s) affected
✓	3	19:35:42	CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR(100), DepartmentID INT)	0 row(s) affected
✓	4	19:36:11	INSERT INTO Employees (EmployeeID, Name, DepartmentID) VALUES (1, 'Riya', 201), (2, 'Pratik', 202), (3, 'Shra...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓	5	19:36:31	CREATE TABLE Departments (DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100))	0 row(s) affected

Step 6: Insert Data into Departments Table

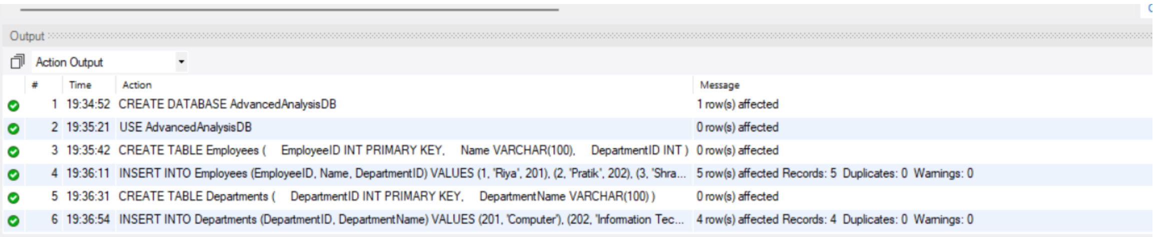
Query:

```
INSERT INTO Departments (DepartmentID, DepartmentName) VALUES
(201, 'Computer'),(202, 'Information Technology'),
(203, 'Electrical'),(205, 'Mechanical');
```

Explanation:

Populates the 'Departments' table with sample data.

Output:



	#	Time	Action	Message
✓	1	19:34:52	CREATE DATABASE AdvancedAnalysisDB	1 row(s) affected
✓	2	19:35:21	USE AdvancedAnalysisDB	0 row(s) affected
✓	3	19:35:42	CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR(100), DepartmentID INT)	0 row(s) affected
✓	4	19:36:11	INSERT INTO Employees (EmployeeID, Name, DepartmentID) VALUES (1, 'Riya', 201), (2, 'Pratik', 202), (3, 'Shra...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓	5	19:36:31	CREATE TABLE Departments (DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100))	0 row(s) affected
✓	6	19:36:54	INSERT INTO Departments (DepartmentID, DepartmentName) VALUES (201, 'Computer'), (202, 'Information Tec...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0

Step 7: Create the Salaries Table

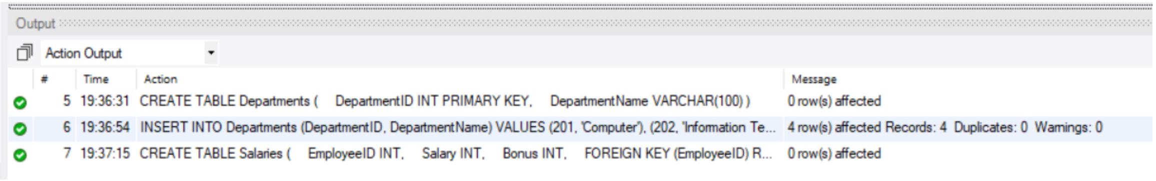
Query:

```
CREATE TABLE Salaries (
    EmployeeID INT,Salary INT,Bonus INT,
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
```

Explanation:

Creates a 'Salaries' table with columns for Employee ID, Salary, and Bonus. It references 'EmployeeID' from the 'Employees' table.

Output:



Output

Action Output

```
 #  Time      Action                                                                                       Message
 5  19:36:31  CREATE TABLE Departments (   DepartmentID INT PRIMARY KEY,   DepartmentName VARCHAR(100) )    0 row(s) affected
 6  19:36:54  INSERT INTO Departments (DepartmentID, DepartmentName) VALUES (201, 'Computer'), (202, 'Information Te...   4 row(s) affected Records: 4  Duplicates: 0  Warnings: 0
 7  19:37:15  CREATE TABLE Salaries (   EmployeeID INT,   Salary INT,   Bonus INT,   FOREIGN KEY (EmployeeID) R...   0 row(s) affected
```

Step 8: Insert Data into Salaries Table

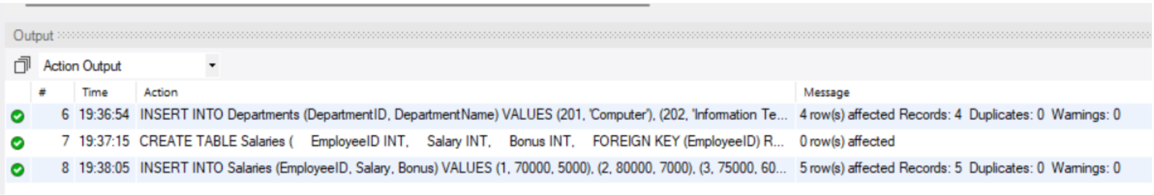
Query:

```
INSERT INTO Salaries (EmployeeID, Salary, Bonus) VALUES
(1, 70000, 5000),(2, 80000, 7000),
(3, 75000, 6000),(4, 72000, 5500),
(5, 68000, 4000);
```

Explanation:

Populates the 'Salaries' table with salary and bonus information.

Output:



Output

Action Output

```
 #  Time      Action                                                                                       Message
 6  19:36:54  INSERT INTO Departments (DepartmentID, DepartmentName) VALUES (201, 'Computer'), (202, 'Information Te...   4 row(s) affected Records: 4  Duplicates: 0  Warnings: 0
 7  19:37:15  CREATE TABLE Salaries (   EmployeeID INT,   Salary INT,   Bonus INT,   FOREIGN KEY (EmployeeID) R...   0 row(s) affected
 8  19:38:05  INSERT INTO Salaries (EmployeeID, Salary, Bonus) VALUES (1, 70000, 5000), (2, 80000, 7000), (3, 75000, 60...   5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0
```

Step 9: Subquery - Employees Earning Above Average Salary

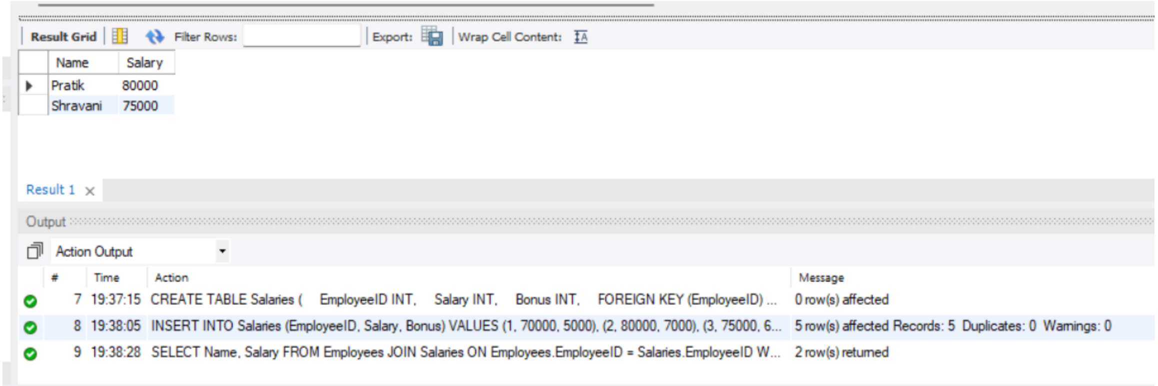
Query:

```
SELECT Name, Salary FROM Employees
JOIN Salaries ON Employees.EmployeeID = Salaries.EmployeeID
WHERE Salary > (SELECT AVG(Salary) FROM Salaries);
```

Explanation:

This query identifies employees whose salary exceeds the average salary.

Output:



Name	Salary
Pratik	80000
Shravani	75000

```
   #  Time      Action                                                                                          Message
   7  19:37:15  CREATE TABLE Salaries (    EmployeeID INT,    Salary INT,    Bonus INT,    FOREIGN KEY (EmployeeID) ...  0 row(s) affected
   8  19:38:05  INSERT INTO Salaries (EmployeeID, Salary, Bonus) VALUES (1, 70000, 5000), (2, 80000, 7000), (3, 75000, 6...  5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0
   9  19:38:28  SELECT Name, Salary FROM Employees JOIN Salaries ON Employees.EmployeeID = Salaries.EmployeeID W...  2 row(s) returned
```

Step 10: CTE - Total Compensation and Rank

Query:

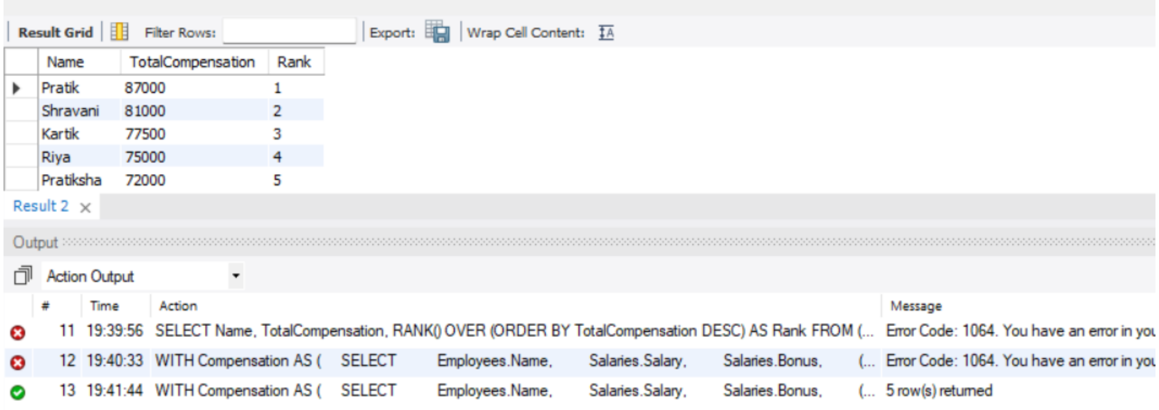
```
WITH Compensation AS (
    SELECT
        Employees.Name,
        Salaries.Salary,  Salaries.Bonus,
        (Salaries.Salary + Salaries.Bonus) AS TotalCompensation
    FROM Employees
    JOIN Salaries ON Employees.EmployeeID = Salaries.EmployeeID
)
```

```
SELECT Name, TotalCompensation, RANK() OVER (ORDER BY TotalCompensation DESC) AS `Rank`

FROM Compensation;Explanation:
```

This query calculates total compensation (salary + bonus) for each employee and ranks them.

Output:



Name	TotalCompensation	Rank
Pratik	87000	1
Shravani	81000	2
Kartik	77500	3
Riya	75000	4
Pratiksha	72000	5

#	Time	Action	Message
11	19:39:56	SELECT Name, TotalCompensation, RANK() OVER (ORDER BY TotalCompensation DESC) AS Rank FROM (...	Error Code: 1064. You have an error in you
12	19:40:33	WITH Compensation AS (SELECT Employees.Name, Salaries.Salary, Salaries.Bonus, (...	Error Code: 1064. You have an error in you
13	19:41:44	WITH Compensation AS (SELECT Employees.Name, Salaries.Salary, Salaries.Bonus, (...	5 row(s) returned

Step 11: Window Function - Average Salary and Deviation

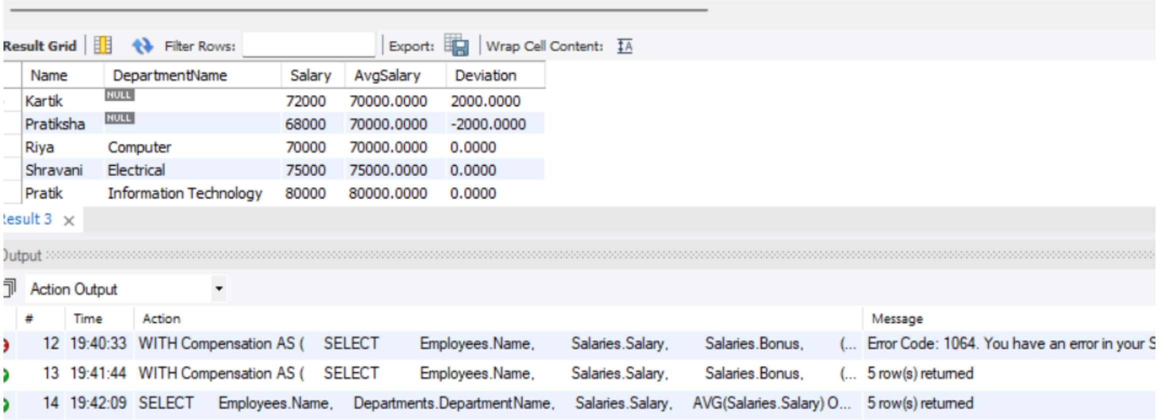
Query:

```
SELECT
   Employees.Name,
   Departments.DepartmentName,
   Salaries.Salary,
   AVG(Salaries.Salary) OVER (PARTITION BY Departments.DepartmentName) AS AvgSalary,
   Salaries.Salary - AVG(Salaries.Salary) OVER (PARTITION BY Departments.DepartmentName) AS Deviation
FROM Employees
LEFT JOIN Salaries ON Employees.EmployeeID = Salaries.EmployeeID
LEFT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

Explanation:

This query calculates the average salary per department and shows how each employee's salary deviates from the average.

Output:



Name	DepartmentName	Salary	AvgSalary	Deviation
Kartik	NULL	72000	70000.0000	2000.0000
Pratiksha	NULL	68000	70000.0000	-2000.0000
Riya	Computer	70000	70000.0000	0.0000
Shravani	Electrical	75000	75000.0000	0.0000
Pratik	Information Technology	80000	80000.0000	0.0000

```
#   Time      Action                                                                                              Message
12  19:40:33  WITH Compensation AS (   SELECT     Employees.Name,    Salaries.Salary,    Salaries.Bonus,    (...  Error Code: 1064. You have an error in your S
13  19:41:44  WITH Compensation AS (   SELECT     Employees.Name,    Salaries.Salary,    Salaries.Bonus,    (...  5 row(s) returned
14  19:42:09  SELECT     Employees.Name,    Departments.DepartmentName,    Salaries.Salary,    AVG(Salaries.Salary) O...  5 row(s) returned
```

Step 12: Drop Tables and Database

Query:

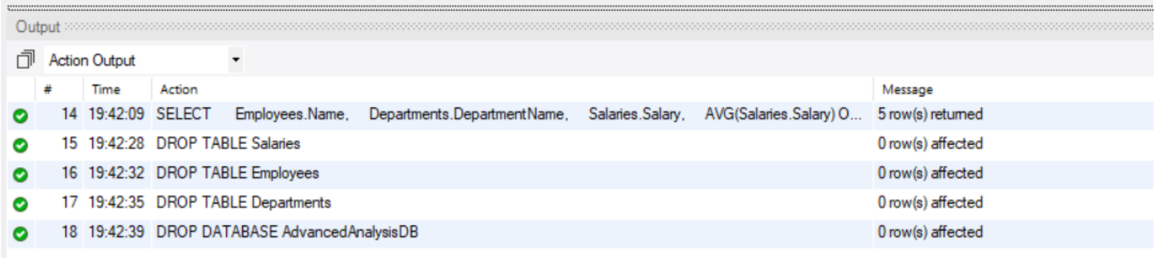
DROP TABLE Salaries;

DROP TABLE Employees;

DROP TABLE Departments;

DROP DATABASE AdvancedAnalysisDB;

Output:



```
#   Time      Action                                                                                              Message
14  19:42:09  SELECT     Employees.Name,    Departments.DepartmentName,    Salaries.Salary,    AVG(Salaries.Salary) O...  5 row(s) returned
15  19:42:28  DROP TABLE Salaries                                                                                 0 row(s) affected
16  19:42:32  DROP TABLE Employees                                                                                0 row(s) affected
17  19:42:35  DROP TABLE Departments                                                                              0 row(s) affected
18  19:42:39  DROP DATABASE AdvancedAnalysisDB                                                                    0 row(s) affected
```