MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

Query 1 | Administration - Data Import/Res...

Limit to 1000 rows

```
#Q1.    Retrieve the names of all states (srcStateName) from the dataset.
SELECT DISTINCT srcStateName
FROM farmersinsurancedata;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName
JAMMU AND KASHMIR
HIMACHAL PRADESH
UTTARAKHAND
HARYANA
GUJARAT
RAJASTHAN
TELANGANA
JHARKHAND
UTTAR PRADESH
TRIPURA

farmersinsurancedata 2 | Read Only

```
#Q2.    Retrieve the total number of farmers covered (TotalFarmersCovered)
#   and the sum insured (SumInsured) for each state (srcStateName),
# ordered by TotalFarmersCovered in descending order.

SELECT
    srcStateName,
    SUM(TotalFarmersCovered) AS TotalFarmersCovered,
    SUM(SumInsured) AS SumInsured
FROM farmersinsurancedata
GROUP BY srcStateName
ORDER BY TotalFarmersCovered DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	TotalFarmersCovered	SumInsured
MADHYA PRADESH	9149678	8413198.288650513
MAHARASHTRA	8939015	4005734.66847685
UTTAR PRADESH	7239576	3298970.927065717
RAJASTHAN	6822958	5363500.742112458
TAMIL NADU	3962760	3025519.4509361982
ODISHA	2586332	1375248.5827995362
HARYANA	2081990	2268958.5814785063

Result 3 | Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```
SELECT *
FROM farmersinsurancedata
WHERE srcYear = 2020;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

rowID	srcYear	srcStateName	srcDistrictName	InsuranceUnits	TotalFarmersCovered	ApplicationsLoaneeFarm
1188	2020	KARNATAKA	Chikkaballapur	16	20	14
1316	2020	KARNATAKA	Chikkamagaluru	124	7	3
2063	2020	ODISHA	Mayurbhanj	9	19	19
2064	2020	ODISHA	Nabarangapur	7	80	98
2065	2020	ODISHA	Nayagarh	7	57	205
2066	2020	ODISHA	Nuapada	21	980	89
2067	2020	ODISHA	Puri	98	9588	29186

```
#Q4.Retrieve all rows where the TotalPopulationRural is
#greater than 1 million and the srcStateName is 'HIMACHAL PRADESH'

SELECT *
FROM farmersinsurancedata
WHERE TotalPopulationRural > 1000000
AND srcStateName = 'HIMACHAL PRADESH';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

rowID	srcYear	srcStateName	srcDistrictName	InsuranceUnits	TotalFarmersCovered	ApplicationsLoane
8	2018	HIMACHAL PRADESH	Kangra	34	30868	31638
1073	2019	HIMACHAL PRADESH	Kangra	34	34564	35348
2263	2020	HIMACHAL PRADESH	Kangra	34	32453	33380
3144	2021	HIMACHAL PRADESH	Kangra	2	11	11
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor

farmersinsurancedata 8 | Apply | Revert

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q5.Retrieve the srcStateName, srcDistrictName, and the sum of FarmersPremiumAmount
#for each district in the year 2018, and display the results ordered by
#FarmersPremiumAmount in ascending order.

SELECT
    srcStateName,
    srcDistrictName,
    SUM(FarmersPremiumAmount) AS FarmersPremiumAmount
FROM farmersinsurancedata
WHERE srcYear = 2018
GROUP BY srcStateName, srcDistrictName
ORDER BY FarmersPremiumAmount ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName	FarmersPremiumAmount
KARNATAKA	Chikkaballapur	0
KARNATAKA	Chikkamagaluru	0
KARNATAKA	Mysuru	0
KARNATAKA	Ramanagara	0
KARNATAKA	Shivamogga	0
KARNATAKA	Tumakuru	0
KARNATAKA	Udupi	0
KARNATAKA	UttarKannada	0

Result 9 | Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q6.Retrieve the total number of farmers covered (TotalFarmersCovered) and the sum of
#premiums (GrossPremiumAmountToBePaid) for each state (srcStateName)
#where the insured land area (InsuredLandArea) is greater than 5.0 and
#the Year is 2018.

SELECT
    srcStateName,
    SUM(TotalFarmersCovered) AS TotalFarmersCovered,
    SUM(GrossPremiumAmountToBePaid) AS TotalPremiumAmount
FROM farmersinsurancedata
WHERE InsuredLandArea > 5.0
AND srcYear = 2018
GROUP BY srcStateName;
```

Result Grid

srcStateName	TotalFarmersCovered	TotalPremiumAmount
JAMMU AND KASHMIR	52842	2472.040069580078
HIMACHAL PRADESH	64766	530.0900039672852
UTTARAKHAND	19726	668.5700073242188
HARYANA	665511	26875.559646606445
RAJASTHAN	1666703	114021.20956420898
TELANGANA	150069	4464.029975891113

Result 10 Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q7.Calculate the average insured land area (InsuredLandArea) for each year (srcYear)


SELECT
    srcYear,
    AVG(InsuredLandArea) AS AvgInsuredLandArea
FROM farmersinsurancedata
GROUP BY srcYear
ORDER BY srcYear;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcYear	AvgInsuredLandArea
2018	38.235249986316866
2019	40.162162148120686
2020	48.35194718522074
2021	39.29111637346095

Result 11 × Read Only

```sql
#Q8.Calculate the total number of farmers covered (TotalFarmersCovered) for
#each district (srcDistrictName)
#where Insurance units is greater than 0.


SELECT
    srcDistrictName,
    SUM(TotalFarmersCovered) AS TotalFarmersCovered
FROM farmersinsurancedata
WHERE InsuranceUnits > 0
GROUP BY srcDistrictName
ORDER BY TotalFarmersCovered DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcDistrictName	TotalFarmersCovered
Bid	1430532
Latur	1184066
Nanded	739712
Osmanabad	700623
Jalna	666086
Ujjain	630787

Result 12 × Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

Q.10

```sql
SELECT
    srcStateName,
    SUM(FarmersPremiumAmount) AS TotalFarmersPremium,
    SUM(StatePremiumAmount) AS TotalStatePremium,
    SUM(GOVPremiumAmount) AS TotalGovPremium,
    SUM(TotalFarmersCovered) AS TotalFarmersCovered
FROM farmersinsurancedata
WHERE SumInsured > 500000
GROUP BY srcStateName
ORDER BY TotalFarmersCovered DESC;
```

Limit to 1000 rows

```sql
#Q10.Retrieve the top 5 districts (srcDistrictName)
#with the highest TotalPopulation in the year 2020.

SELECT
    srcDistrictName,
    TotalPopulation
FROM farmersinsurancedata
WHERE srcYear = 2020
ORDER BY TotalPopulation DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

srcDistrictName	TotalPopulation
Pune	9429408
Thane	8070032
Jaipur	6626178
Nashik	6107187
Allahabad	5954391

farmersinsurancedata 14 × Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q11.Retrieve the srcStateName, srcDistrictName, and SumInsured for the 10
#districts with the lowest non-zero FarmersPremiumAmount,
#ordered by insured sum and then the FarmersPremiumAmount.

SELECT
    srcStateName,
    srcDistrictName,
    SumInsured,
    FarmersPremiumAmount
FROM farmersinsurancedata
WHERE FarmersPremiumAmount > 0
ORDER BY SumInsured ASC, FarmersPremiumAmount ASC
LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

srcStateName	srcDistrictName	SumInsured	FarmersPremiumAmount
KARNATAKA	Kalaburgi	0.0044	0.0001
KARNATAKA	Kolar	0.0048	0.0001
KARNATAKA	Davangere	0.0051	0.0001
KARNATAKA	Ballari	0.0052	0.0001
KARNATAKA	Mandya	0.0056	0.0001

farmersinsurancedata 15 | Read Only

Limit to 1000 rows

```sql
#Q12.Retrieve the top 3 states (srcStateName) along with the year (srcYear) where
#the ratio of insured farmers (TotalFarmersCovered) to the total population (TotalPopulation)
#is highest. Sort the results by the ratio in descending order.

SELECT
    srcStateName,
    srcYear,
    (SUM(TotalFarmersCovered) * 1.0 / SUM(TotalPopulation)) AS InsuredFarmersRatio
FROM farmersinsurancedata
WHERE TotalPopulation > 0  -- To avoid division by zero
GROUP BY srcStateName, srcYear
ORDER BY InsuredFarmersRatio DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

srcStateName	srcYear	InsuredFarmersRatio
CHHATTISGARH	2021	0.04981
TRIPURA	2020	0.04683
TRIPURA	2021	0.04637

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q13.Create StateShortName by retrieving the
#first 3 characters of the srcStateName for each unique state.

SELECT
    DISTINCT srcStateName,
    LEFT(srcStateName, 3) AS StateShortName
FROM farmersinsurancedata;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	StateShortName
JAMMU AND KASHMIR	JAM
HIMACHAL PRADESH	HIM
UTTARAKHAND	UTT
HARYANA	HAR
GUJARAT	GUJ

```sql
#Q15.Retrieve the srcDistrictName where the district name starts with 'B'

SELECT DISTINCT srcDistrictName
FROM farmersinsurancedata
WHERE srcDistrictName LIKE 'B%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcDistrictName
Bilaspur
Bageshwar
Bhiwani
Banswara
Baran

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q15.Retrieve the srcStateName and srcDistrictName where the
# district name contains the word 'pur' at the end

SELECT DISTINCT srcStateName, srcDistrictName
FROM farmersinsurancedata
WHERE srcDistrictName LIKE '%pur';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName
JAMMU AND KASHMIR	Udhampur
HIMACHAL PRADESH	Bilaspur
HIMACHAL PRADESH	Hamirpur
RAJASTHAN	Bharatpur
RAJASTHAN	Dhaulpur

```sql
#Q16.Perform an INNER JOIN between the srcStateName and srcDistrictName columns to
#retrieve the aggregated FarmersPremiumAmount for districts where the
#district's Insurance units for an individual year are greater than 10
SELECT
    f1.srcStateName,
    f1.srcDistrictName,
    SUM(f1.FarmersPremiumAmount) AS TotalFarmersPremium
FROM farmersinsurancedata f1
INNER JOIN farmersinsurancedata f2
    ON f1.srcStateName = f2.srcStateName
    AND f1.srcDistrictName = f2.srcDistrictName
WHERE f2.InsuranceUnits > 10
GROUP BY f1.srcStateName, f1.srcDistrictName
ORDER BY TotalFarmersPremium DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName	TotalFarmersPremium
MAHARASHTRA	Bid	58115.43896484375
MADHYA PRADESH	Ujjain	49540.92138671875
MAHARASHTRA	Latur	46801.279296875
MADHYA PRADESH	Rajgarh	37879.83984375
MADHYA PRADESH	Sehore	37013.95947265625

Result 20 × Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q.17
SELECT
    srcStateName,
    srcDistrictName,
    srcYear AS Year,
    TotalPopulation,
    MAX(FarmersPremiumAmount) AS HighestFarmersPremiumAmount
FROM FarmersInsuranceData
GROUP BY srcStateName, srcDistrictName, srcYear, TotalPopulation
HAVING MAX(FarmersPremiumAmount) > 200000000  -- 20 Crores
ORDER BY HighestFarmersPremiumAmount DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName	Year	TotalPopulation	HighestFarmersPremiumAmount

Since dataset does not contain FarmersPremiumAmount values anywhere near 20 crores (200,000,000), this query will likely return no results.

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q.18
SELECT
    f.srcStateName,
    f.srcDistrictName,
    SUM(f.FarmersPremiumAmount) AS TotalFarmersPremiumAmount,
    AVG(f.TotalPopulation) AS AvgPopulation
FROM FarmersInsuranceData f
LEFT JOIN (
    SELECT srcStateName, srcDistrictName, TotalPopulation
    FROM FarmersInsuranceData
) p ON f.srcStateName = p.srcStateName AND f.srcDistrictName = p.srcDistrictName
GROUP BY f.srcStateName, f.srcDistrictName
HAVING SUM(f.FarmersPremiumAmount) > 1000000000  -- 100 Crores
ORDER BY TotalFarmersPremiumAmount DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName	TotalFarmersPremiumAmount	AvgPopulation

Result 57 × Read Only

Output

Since no districts meet the 100 crore (1,000,000,000**)** threshold in the dataset, the query returns no results.

```sql
#Q19.Write a query to find the districts (srcDistrictName) where the TotalFarmersCovered is
#greater than the average TotalFarmersCovered across all records.

SELECT DISTINCT srcDistrictName, srcStateName, TotalFarmersCovered
FROM farmersinsurancedata
WHERE TotalFarmersCovered > (
    SELECT AVG(TotalFarmersCovered) FROM farmersinsurancedata
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcDistrictName	srcStateName	TotalFarmersCovered
Kangra	HIMACHAL PRADESH	30868
Bhiwani	HARYANA	43225
Fatehabad	HARYANA	51867
Hisar	HARYANA	85052
Jind	HARYANA	53620

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q20.
SELECT DISTINCT srcStateName
FROM FarmersInsuranceData
WHERE SumInsured > (
    SELECT SumInsured
    FROM FarmersInsuranceData
    WHERE FarmersPremiumAmount = (SELECT MAX(FarmersPremiumAmount) FROM FarmersInsuranceData)
    LIMIT 1
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName

FarmersInsuranceData 61 × Read Only

##From your dataset, the highest FarmersPremiumAmount is ₹7244.42 (for the district Bid, Maharashtra), and its SumInsured is ₹275,019.

##Since the SumInsured threshold is very low, it's highly likely that many states do not exceed this value, which is why the query returned no results.

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q21.Write a query to find the srcDistrictName where the FarmersPremiumAmount is higher than
#the average FarmersPremiumAmount of the state that has the highest TotalPopulation.
SELECT DISTINCT srcDistrictName, srcStateName, FarmersPremiumAmount
FROM farmersinsurancedata
WHERE FarmersPremiumAmount > (
    -- Subquery to get the average FarmersPremiumAmount of the state with the highest TotalPopu
    SELECT AVG(FarmersPremiumAmount)
    FROM farmersinsurancedata
    WHERE srcStateName = (
        -- Subquery to get the state with the highest TotalPopulation
        SELECT srcStateName
        FROM farmersinsurancedata
        ORDER BY TotalPopulation DESC
        LIMIT 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcDistrictName	srcStateName	FarmersPremiumAmount
Bhiwani	HARYANA	702.08
Fatehabad	HARYANA	1001.75
Hisar	HARYANA	1355.27
Jind	HARYANA	895.54
Kaithal	HARYANA	636.37

farmersinsurancedata 42 | Read Only

```sql
#Q22.Use the ROW_NUMBER() function to assign a row number to each record in the dataset
#ordered by total farmers covered in descending order.

SELECT
    ROW_NUMBER() OVER (ORDER BY TotalFarmersCovered DESC) AS RowNum,
    srcStateName,
    srcDistrictName,
    srcYear,
    TotalFarmersCovered
FROM farmersinsurancedata;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RowNum	srcStateName	srcDistrictName	srcYear	TotalFarmersCovered
1	MAHARASHTRA	Bid	2019	548572
2	MAHARASHTRA	Nanded	2021	426801
3	MAHARASHTRA	Latur	2019	407452
4	MAHARASHTRA	Bid	2018	387806
5	MAHARASHTRA	Latur	2021	367746

Result 43 | Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q23.Use the RANK() function to rank the districts (srcDistrictName) based on the SumInsured
#(descending) and partition by alphabetical srcStateName.

SELECT
    srcStateName,
    srcDistrictName,
    SumInsured,
    RANK() OVER (PARTITION BY srcStateName ORDER BY SumInsured DESC) AS RankInState
FROM farmersinsurancedata
ORDER BY srcStateName ASC, RankInState;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName	SumInsured	RankInState
ANDHRA PRADESH	Kurnool	191969	1
ANDHRA PRADESH	Y.S.R.	177442	2
ANDHRA PRADESH	Srikakulam	149882	3
ANDHRA PRADESH	Prakasam	115235	4
ANDHRA PRADESH	Krishna	110285	5

Result 44 × Read Only

```sql
#Q24.Use the SUM() window function to calculate a cumulative sum of FarmersPremiumAmount
#for each district (srcDistrictName), ordered ascending by the srcYear, partitioned by
#srcStateName.
SELECT
    srcStateName,
    srcDistrictName,
    srcYear,
    FarmersPremiumAmount,
    SUM(FarmersPremiumAmount) OVER (
        PARTITION BY srcStateName, srcDistrictName
        ORDER BY srcYear ASC
    ) AS CumulativePremium
FROM farmersinsurancedata
ORDER BY srcStateName, srcDistrictName, srcYear;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

srcStateName	srcDistrictName	srcYear	FarmersPremiumAmount	CumulativePremium
ANDHRA PRADESH	Anantapur	2018	541.83	541.8300170898438
ANDHRA PRADESH	Anantapur	2019	0.189	542.019017085433
ANDHRA PRADESH	Chittoor	2018	99.65	99.6500015258789
ANDHRA PRADESH	Chittoor	2019	0.4411	100.09110152721405
ANDHRA PRADESH	East Godavari	2018	102.6	102.5999984741211

Result 46 × Read Only

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q25.Create a table 'districts' with DistrictCode as the primary key and columns for
#DistrictName and StateCode.Create another table 'states' with StateCode as primary key
#and column for StateName.
-- Creating the 'states' table
CREATE TABLE states (
    StateCode VARCHAR(10) PRIMARY KEY,  -- Unique code for each state
    StateName VARCHAR(255) NOT NULL     -- Name of the state
);

-- Creating the 'districts' table
CREATE TABLE districts (
    DistrictCode VARCHAR(10) PRIMARY KEY,  -- Unique code for each district
    DistrictName VARCHAR(255) NOT NULL,    -- Name of the district
    StateCode VARCHAR(10) NOT NULL,        -- Foreign key referencing 'states' table
    FOREIGN KEY (StateCode) REFERENCES states(StateCode) ON DELETE CASCADE
);
```

Output

Action Output

#	Time	Action	Message
75	22:11:33	SELECT srcStateName, srcDistrictName, srcYear, FarmersPre...	1820 row(s) returned
76	22:15:30	CREATE TABLE states (StateCode VARCHAR(10) PRIMARY KEY, -- ...	0 row(s) affected
77	22:15:31	CREATE TABLE districts (DistrictCode VARCHAR(10) PRIMARY KEY, ...	0 row(s) affected

```sql
#Q26.Add a foreign key constraint to the districts table that references the StateCode
#column from a states table.

ALTER TABLE districts
ADD CONSTRAINT fk_statecode
FOREIGN KEY (StateCode)
REFERENCES states(StateCode)
ON DELETE CASCADE;
```

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q27.Update the FarmersPremiumAmount to 500.0 for the record where rowID is 1.

UPDATE farmersinsurancedata
SET FarmersPremiumAmount = 500.0
WHERE rowID = 1;
```

Output

Action Output

#	Time	Action	Message
1	22:21:08	UPDATE farmersinsurancedata SET FarmersPremiumAmount = 500.0 WHER...	0 row(s) affected Rows matched: 0 Changed: 0

```sql
#Q28.Update the Year to '2021' for all records where srcStateName is 'HIMACHAL PRADESH'

UPDATE farmersinsurancedata
SET srcYear = 2021
WHERE srcStateName = 'HIMACHAL PRADESH';
```

Output

Action Output

#	Time	Action	Message
8	22:24:13	UPDATE farmersinsurancedata SET srcYear = 2021 WHERE srcStateNam...	Error Code: 1175. You are using safe update mode ar
9	22:24:51	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
10	22:25:03	UPDATE farmersinsurancedata SET srcYear = 2021 WHERE srcStateNa...	30 row(s) affected Rows matched: 40 Changed: 30

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

```sql
#Q29.Delete all records where the TotalFarmersCovered is less than 10000 and Year is 2020.

DELETE FROM farmersinsurancedata
WHERE TotalFarmersCovered < 10000
AND srcYear = 2020;
```

Output

Action Output

#	Time	Action	Message
29	23:19:46	SELECT DISTINCT srcStateName FROM FarmersInsuranceData WHERE ...	0 row(s) returned
30	08:06:36	DELETE FROM farmersinsurancedata WHERE TotalFarmersCovered < 10...	0 row(s) affected
31	08:06:38	DELETE FROM farmersinsurancedata WHERE TotalFarmersCovered < 10...	0 row(s) affected

MySQL - Farmers Insurance Analysis - Assignment
Group - Titans

Executive Summary

The analysis of the farmers' insurance dataset reveals key insights into premium payments, insured amounts, and coverage across states and districts.

1. **Low Premium Contributions:**
   - The highest recorded FarmersPremiumAmount is only ₹7244.42 in Bid, Maharashtra, indicating minimal premium contributions. This suggests limited participation in premium-based insurance schemes.
2. **Moderate Insured Land Coverage:**
   - The SumInsured for Bid, Maharashtra, is ₹275,019, which is relatively low. No states surpass this insured amount, indicating that large-scale agricultural coverage is lacking.
3. **No Districts Meet High Premium Thresholds:**
   - The dataset lacks districts with FarmersPremiumAmount exceeding ₹20 crores, and the total sum across all records is just ₹5.62 lakh, far below expected large-scale agricultural insurance values.
4. **Limited State-Level Insurance Exceeding District Maximums:**
   - No states reported a SumInsured higher than the district with the highest FarmersPremiumAmount, implying that district-level insured sums are not significantly surpassed at the state level.

Conclusion & Recommendations:

- The low insurance adoption rate suggests a need for greater awareness and participation incentives.
- State-level intervention is necessary to boost farmer participation in premium-based schemes.
- Future investigations should explore policy effectiveness**,** claim settlements**,** and subsidy impacts to understand barriers to insurance adoption.