AI-Powered Text-to-SQL Generator 🚀

Enter your natural language query:

take sum of sales_revenue_sum column by each year from the fact_ad_summary_ui table

Generate SQL

Generated SQL Query:

```sql
SELECT
  SUM(T1.SALES_REVENUE_SUM),
  T2.YEAR
FROM FACT_AD_SUMMARY_UI AS T1
JOIN DIM_DATE AS T2
  ON T1.DATE = T2.DAY
GROUP BY
  T2.YEAR;
```

Optimized SQL Query:

```sql
SELECT
  SUM(T1.SALES_REVENUE_SUM),
  T2.YEAR
FROM FACT_AD_SUMMARY_UI AS T1
JOIN DIM_DATE AS T2
  ON T1.DATE = T2.DAY
GROUP BY
  T2.YEAR;
```

```
ORDER BY T1.SALES_REVENUE_SUM
```

Run Query

Executing SQL Query:

```
SELECT
  SUM(T1.SALES_REVENUE_SUM),
  T2.YEAR
FROM FACT_AD_SUMMARY_UI AS T1
JOIN DIM_DATE AS T2
  ON T1.DATE = T2.DAY
GROUP BY
  T2.YEAR
```

Query Results:

	SUM(T1.SALES_REVENUE_SUM)	YEAR
0	1,770,225,884.5392	2023-01-01
1	868,842,932.7972	2023-12-31

Performance Insights:

	step	id	parentOperators	operation	objects	alias	expressions
0	None	None	None	GlobalStats	None	None	None
1	1	0	None	Result	None	None	SUM(T1.SALES_REVENUE_SUM), T2.YEAR
2	1	1	[0]	Aggregate	None	None	aggExprs: [SUM(T1.SALES_REVENUE_SUM
3	1	2	[1]	InnerJoin	None	None	joinKey: (T2.DAY = T1.DATE)
4	1	3	[2]	TableScan	PRD_PLA	T2	DAY, YEAR
5	1	4	[2]	JoinFilter	None	None	joinKey: (T2.DAY = T1.DATE)
6	1	5	[4]	TableScan	PRD_PLA	T1	DATE, SALES_REVENUE_SUM