

Name: Prathima s

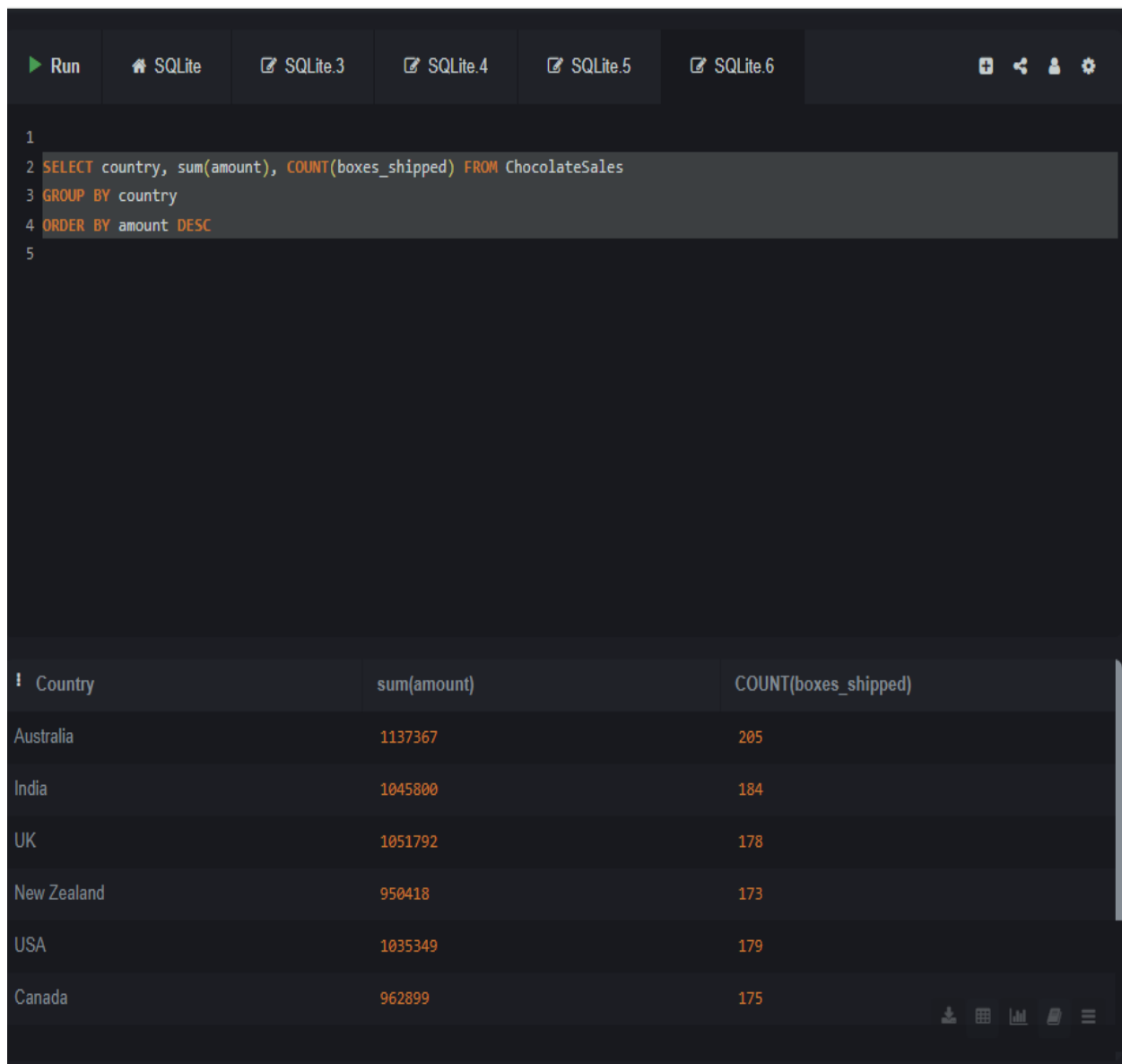
Email- prathuprathu29@gmail.com

Course name: SQL

Database-

Chocolatesales(<https://www.kaggle.com/datasets/atharvasoundankar/chocolate-sales>)

Q1. Which country gives the maximum sales and boxes_shipped



The screenshot shows a SQL query editor with a dark theme. The query is as follows:

```
1  
2 SELECT country, sum(amount), COUNT(boxes_shipped) FROM ChocolateSales  
3 GROUP BY country  
4 ORDER BY amount DESC  
5
```

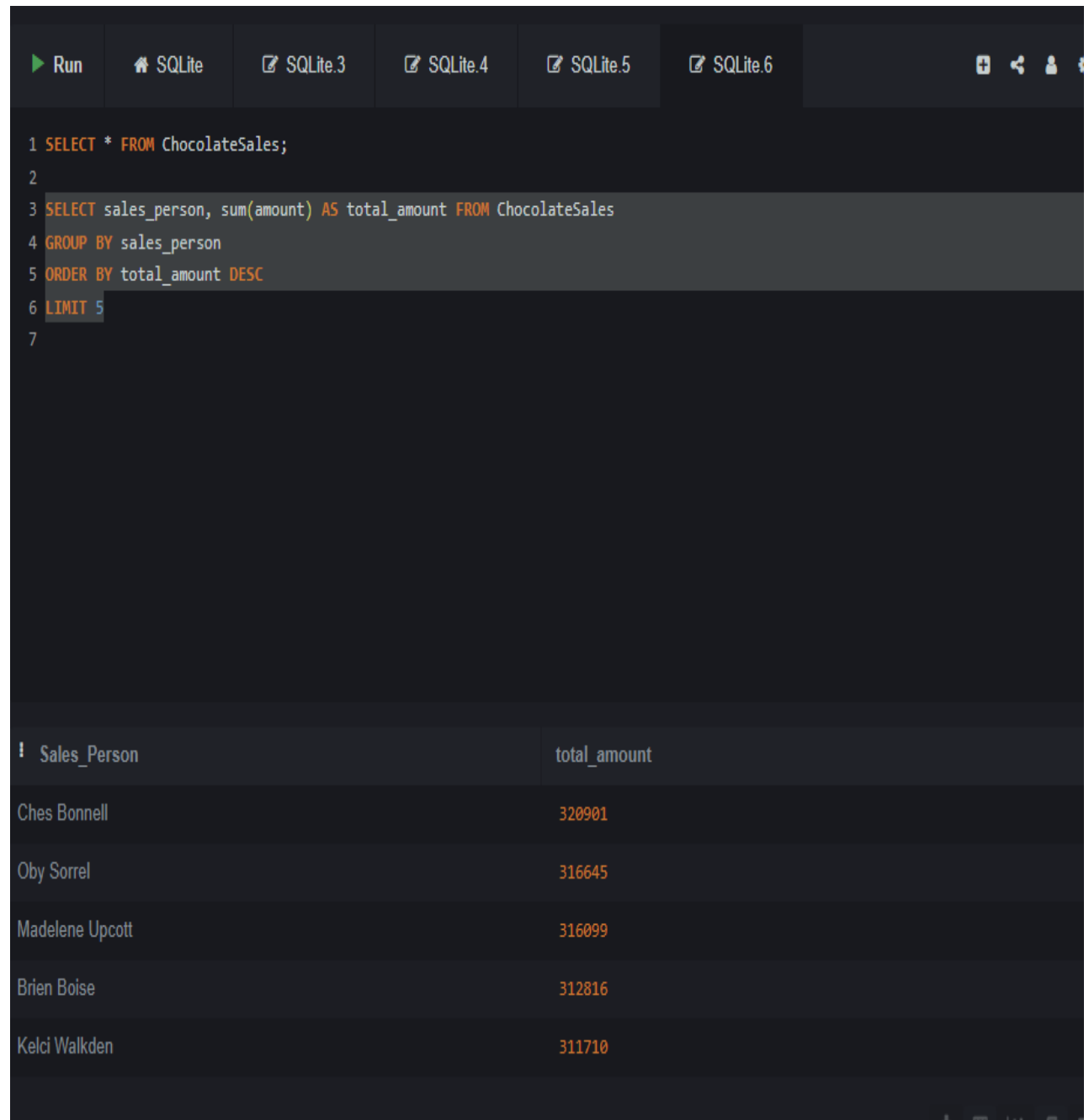
Below the query editor, the results are displayed in a table with three columns: Country, sum(amount), and COUNT(boxes_shipped). The results are sorted by sum(amount) in descending order.

Country	sum(amount)	COUNT(boxes_shipped)
Australia	1137367	205
India	1045800	184
UK	1051792	178
New Zealand	950418	173
USA	1035349	179
Canada	962899	175

Q2. Top 5 salesperson with highest amount?

SELECT sales_person, sum(amount) as total_amount from ChocolateSales

GROUP by sales_person order by total_amount DESC LIMIT 5



The screenshot shows a SQLite IDE interface. At the top, there are tabs for 'Run', 'SQLite', and several 'SQLite.3' through 'SQLite.6' files. The main area displays a SQL query with line numbers 1 through 7. The query is:
1 SELECT * FROM ChocolateSales;
2
3 SELECT sales_person, sum(amount) AS total_amount FROM ChocolateSales
4 GROUP BY sales_person
5 ORDER BY total_amount DESC
6 LIMIT 5
7
Below the query editor, the results are displayed in a table with two columns: 'Sales_Person' and 'total_amount'. The table contains five rows of data, with the 'total_amount' values highlighted in orange.

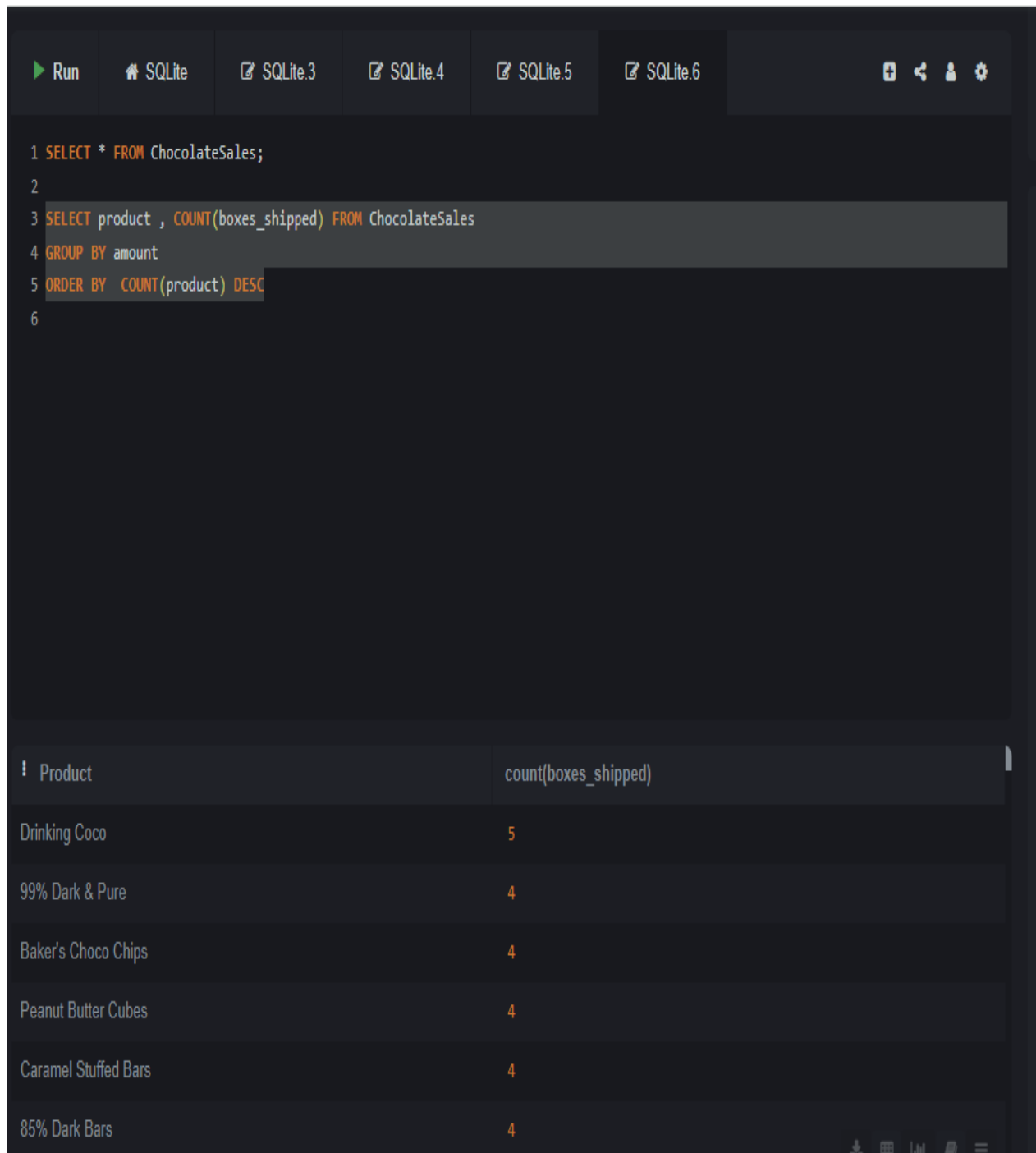
Sales_Person	total_amount
Ches Bonnell	320901
Oby Sorrel	316645
Madelene Upcott	316099
Brien Boise	312816
Kelci Walkden	311710

Q3. Which product is most shipped?

SELECT product , count(boxes_shipped) from ChocolateSales

group by amount

order by count(product) DESC



The screenshot shows a SQL IDE interface with a dark theme. At the top, there is a toolbar with a 'Run' button and several database connection tabs labeled 'SQLite', 'SQLite.3', 'SQLite.4', 'SQLite.5', and 'SQLite.6'. Below the toolbar, a SQL query is entered in the editor:

```
1 SELECT * FROM ChocolateSales;
2
3 SELECT product , COUNT(boxes_shipped) FROM ChocolateSales
4 GROUP BY amount
5 ORDER BY COUNT(product) DESC
6
```

Below the query editor, the results of the query are displayed in a table. The table has two columns: 'Product' and 'count(boxes_shipped)'. The results are as follows:

Product	count(boxes_shipped)
Drinking Coco	5
99% Dark & Pure	4
Baker's Choco Chips	4
Peanut Butter Cubes	4
Caramel Stuffed Bars	4
85% Dark Bars	4

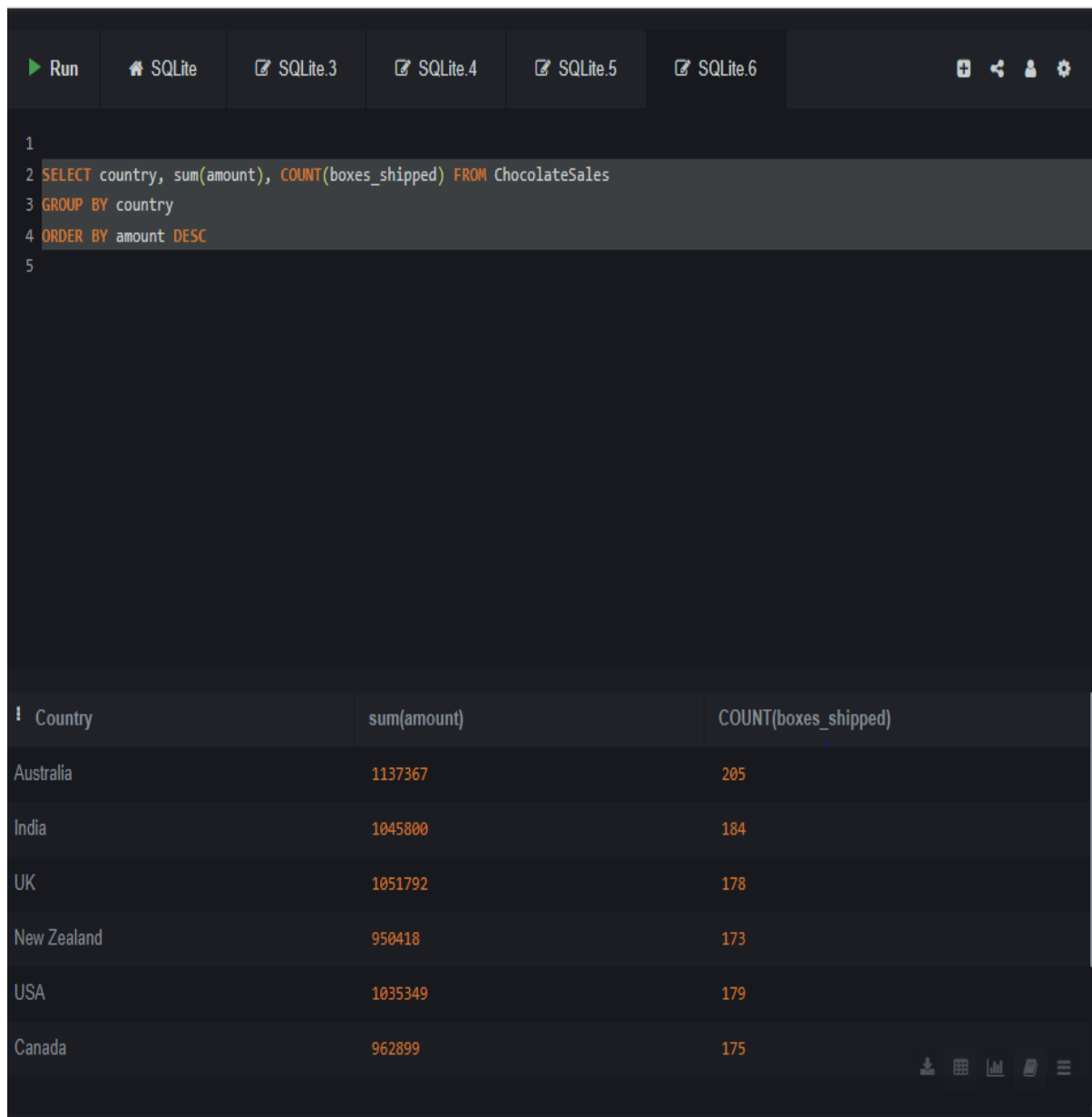
Q4. Best-selling product by total amount sold?

SELECT product, sum(amount) as total_amt FROM ChocolateSales

GROUP by product

ORDER by total_amt desc

LIMIT 10



The screenshot shows a SQL IDE interface with a dark theme. At the top, there is a toolbar with buttons for 'Run', 'SQLite', 'SQLite.3', 'SQLite.4', 'SQLite.5', and 'SQLite.6'. Below the toolbar, a SQL query is entered in a text area:

```
1  
2 SELECT country, sum(amount), COUNT(boxes_shipped) FROM ChocolateSales  
3 GROUP BY country  
4 ORDER BY amount DESC  
5
```

Below the query editor, the results of the query are displayed in a table with three columns: 'Country', 'sum(amount)', and 'COUNT(boxes_shipped)'. The table contains six rows of data, sorted by 'sum(amount)' in descending order.

Country	sum(amount)	COUNT(boxes_shipped)
Australia	1137367	205
India	1045800	184
UK	1051792	178
New Zealand	950418	173
USA	1035349	179
Canada	962899	175

Q4. Count of box shipped for each salesperson?

SELECT sales_person,count(boxes_shipped) as total_box_shipped from
ChocolateSales GROUP by sales_person ORDER by total_box_shipped DESC

RunSQLiteSQLite.3SQLite.4SQLite.5SQLite.6

```
1 SELECT sales_person,COUNT(boxes_shipped) AS total_box_shipped FROM ChocolateSales
2 GROUP BY sales_person
3 ORDER BY total_box_shipped DESC
4
5
6
```

Sales_Person	total_box_shipped
Kelci Walkden	54
Brien Boise	53
Van Tuxwell	51
Beverie Moffet	50
Oby Sorrel	49
Dennison Crosswaite	49
Ches Bonnell	48

Q5. Average sales amt per box shipped?

```
SELECT product,avg(amount/boxes_shipped) as Avg_sales_per_box FROM
ChocolateSales GROUP by sales_person ORDER by Avg_sales_per_box DESC
```

RunSQLiteSQLite.3SQLite.4SQLite.5SQLite.6

```
1 SELECT product,avg(amount/boxes_shipped) AS Avg_sales_per_box FROM ChocolateSales
2 GROUP BY sales_person
3 ORDER BY Avg_sales_per_box DESC
4
5
```

! Product	Avg_sales_per_box
Orange Choco	177.67441860465115
Eclairs	174.2439024390244
85% Dark Bars	172.15686274509804
Almond Choco	149.24444444444444
Peanut Butter Cubes	136.57777777777778
Smooth Silky Salty	114.53488372093024
85% Dark Bars	113.6304347826087

Q6. Sales performance of salesperson by country?

```
SELECT sales_person,country, sum(amount) as total_amt FROM ChocolateSales
GROUP by sales_person, country order by total_amt DESC
```

Run	SQLite	SQLite.3	SQLite.4	SQLite.5	SQLite.6	
-----	--------	----------	----------	----------	----------	--

```

1 SELECT sales_person, country, sum(amount) AS total_amt FROM ChocolateSales
2 GROUP BY sales_person, country
3 ORDER BY total_amt DESC

```

Sales_Person	Country	total_amt
Ches Bonnell	India	98763
Van Tuxwell	India	85785
Oby Sorrel	India	75579
Beverie Moffet	Canada	71995
Madelene Upcott	UK	71330
Mallorie Waber	Australia	71134
Jehu Rudeforth	UK	69895

2) Database- Financial risk compliance

<https://www.kaggle.com/datasets/atharvasoundankar/big-4-financial-risk-insights-2020-2025>

Q1. What is a max, min, avg total_revenue_impact on each firm?

```
SELECT firm_name, max(total_revenue_impact), min(total_revenue_impact),
round(avg(total_revenue_impact),2) as avg_revenue FROM
big4_financial_risk_compliance GROUP by firm_name ORDER by avg_revenue DESC
```



The screenshot shows a SQL query editor with a toolbar at the top containing buttons for 'Run', 'SQLite', and several 'SQLite' tabs (3 through 7). The query is entered in the main text area and is as follows:

```
1 SELECT firm_name, max(total_revenue_impact), min(total_revenue_impact),
2 round(avg(total_revenue_impact), 2) AS avg_revenue FROM big4_financial_risk_compliance
3 GROUP BY firm_name
4 ORDER BY avg_revenue DESC
```

Below the query editor, the results are displayed in a table with the following columns: Firm_Name, max(total_revenue_impact), min(total_revenue_impact), and avg_revenue. The results are sorted by avg_revenue in descending order.

Firm_Name	max(total_revenue_impact)	min(total_revenue_impact)	avg_revenue
Ernst & Young	497.06	61.17	298.39
Deloitte	495.19	54.07	284.91
KPMG	478	65	260.69
PwC	485.64	33.46	244.34

Q2. Total revenue impact by industry?

```
SELECT industry_affected, round(sum(total_revenue_impact),2) as total_revenue FROM
big4_financial_risk_compliance GROUP by industry_affected ORDER by total_revenue
DESC
```


Run

SQLite

SQLite.3

SQLite.4

SQLite.5

SQLite.6

SQLite.7

```
1 SELECT industry_affected, round(sum(total_revenue_impact),2)AS total_revenue FROM big4_financial_risk_compliance
2 GROUP BY industry_affected
3 ORDER BY total_revenue DESC
```

Industry_Affected	total_revenue
Retail	8696.3
Tech	7141.81
Finance	5940.2
Healthcare	5475.59

Q3. Firms with total audit engagements higher than the average?

```
SELECT      firm_name,round(avg(total_audit_engagements),2)      as
Avg_total_audit_engagements  FROM  big4_financial_risk_compliance  GROUP  by
firm_name      HAVING      round(avg(total_audit_engagements),2)  >  (SELECT
round(avg(total_audit_engagements),2) from big4_financial_risk_compliance)
```

Run SQLite SQLite.3 SQLite.4 SQLite.5 SQLite.6 SQLite.7

```

1 SELECT firm_name,round(avg(total_audit_engagements),2) AS Avg_total_audit_engagements FROM big4_financial_risk_compliance
2 GROUP BY firm_name HAVING round(avg(total_audit_engagements),2) > (SELECT round(avg(total_audit_engagements),2) FROM big4_finan
3

```

Firm_Name	Avg_total_audit_engagements
KPMG	3052.5
PwC	3197.44

Q4. Industries with more compliance violation than average?

```

SELECT industry_affected ,sum(compliance_violations) as
total_violation,round(avg(compliance_violations),2) FROM
big4_financial_risk_compliance GROUP by industry_affected HAVING
sum(compliance_violations)>(SELECT round(avg(compliance_violations),2) from
big4_financial_risk_compliance)

```

Run

SQLite

SQLite.3

SQLite.4

SQLite.5

SQLite.6

SQLite.7

```
1 SELECT industry_affected ,sum(compliance_violations) AS total_violation,round(avg(compliance_violations),2) AS avg_voilation
2 FROM big4_financial_risk_compliance
3 GROUP BY industry_affected HAVING sum(compliance_violations)>(SELECT round(avg(compliance_violations),2)
4 FROM big4_financial_risk_compliance)
```

Industry_Affected	total_violation	avg_voilation
Finance	1669	83.45
Healthcare	2665	111.04
Retail	3226	119.48
Tech	2988	103.03

Q5. Average employee workload for each firm?

```
SELECT firm_name, round(avg(employee_workload),2) as avg_workload FROM
big4_financial_risk_compliance GROUP by firm_name
```

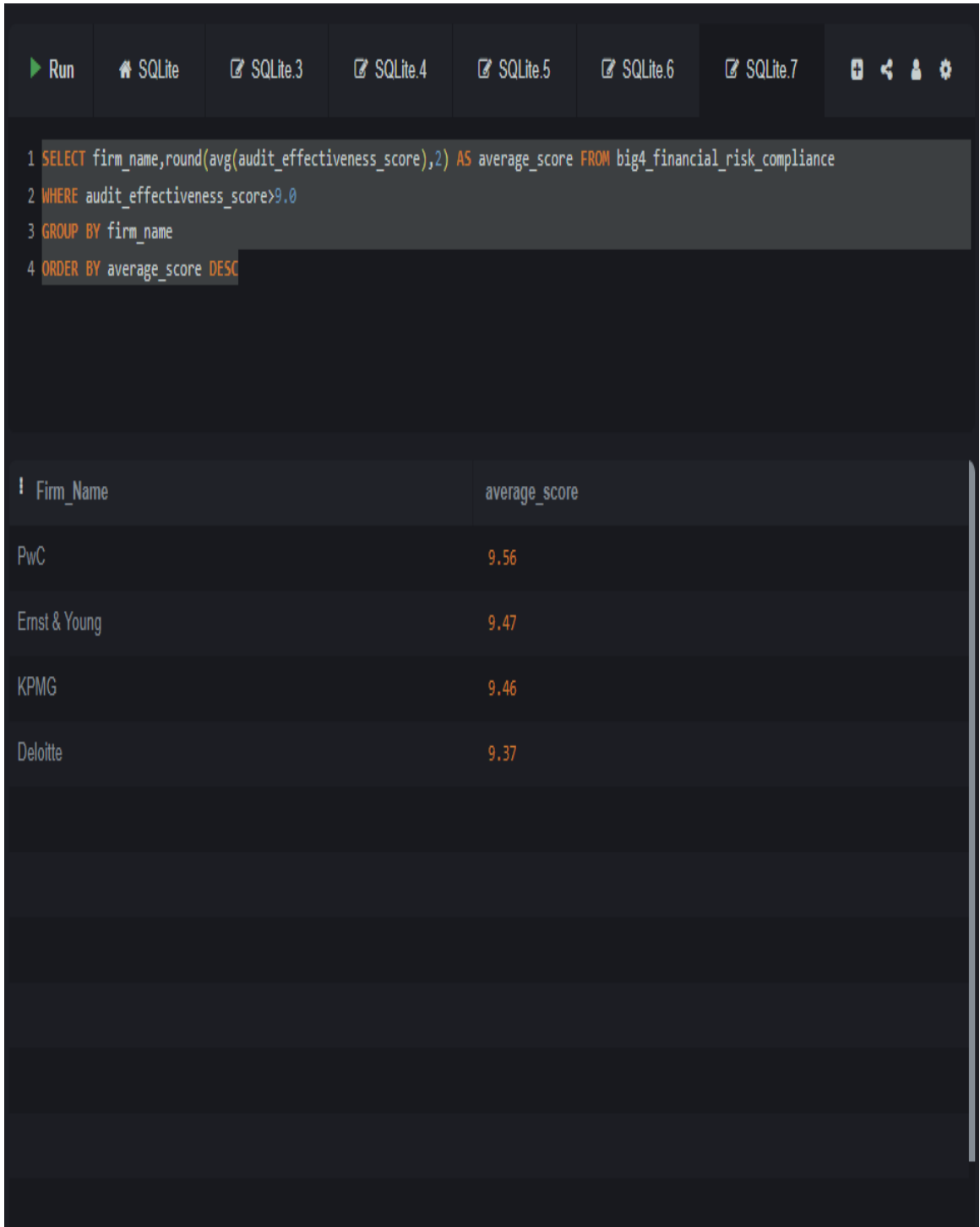
RunSQLiteSQLite.3SQLite.4SQLite.5SQLite.6SQLite.7

```
1 SELECT firm_name, round(avg(employee_workload),2) AS avg_workload FROM big4_financial_risk_compliance
2 GROUP BY firm_name
3 |
```

Firm_Name	avg_workload
Deloitte	62.43
Ernst & Young	60.91
KPMG	59.23
PwC	57.92

Q6. Firms with audit effectiveness score above a threshold (e.g., 9.0)?

```
SELECT firm_name,round(avg(audit_effectiveness_score),2) as average_score from
big4_financial_risk_compliance WHERE audit_effectiveness_score>9.0 GROUP by
firm_name ORDER by average_score DESC
```



Q7. Total fraud cases detected by industry?

```
SELECT industry_affected, sum(fraud_cases_detected) as Total_fraud_cases FROM
big4_financial_risk_compliance GROUP by industry_affected ORDER by
Total_fraud_cases DESC
```

RunSQLite.SQLite.3.SQLite.4.SQLite.5.SQLite.6.SQLite.7

```
1 SELECT industry_affected, sum(fraud_cases_detected) AS Total_fraud_cases FROM big4_financial_risk_compliance
2 GROUP BY industry_affected
3 ORDER BY Total_fraud_cases DESC
```

Industry_Affected	Total_fraud_cases
Healthcare	1410
Tech	1394
Retail	1365
Finance	1101

Q8. Trend analysis over time(year)

```
SELECT    year,      sum(total_audit_engagements)      as      TotalEngagements,
sum(high_risk_cases) as TotalHighRisk, sum(compliance_violations) as TotalViolation ,
sum(fraud_cases_detected) as TotalFraud from big4_financial_risk_compliance GROUP
by year
```

Run	SQLite	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7				
-----	--------	----------	----------	----------	----------	----------	--	--	--	--

```

1 SELECT year , sum(total_audit_engagements) AS TotalEngagements, sum(high_risk_cases) AS TotalHighRisk,
2 sum(compliance_violations) AS TotalViolation , sum(fraud_cases_detected) AS TotalFraud FROM big4_financial_risk_compliance
3 GROUP BY year
4

```

Year	TotalEngagements	TotalHighRisk	TotalViolation	TotalFraud
2020	55139	6063	2441	1042
2021	51888	4684	1562	963
2022	43667	4195	1556	667
2023	35752	3006	1454	727
2024	43753	5513	1529	938
2025	48253	4312	2006	933

Q9. Firm comparison by client satisfaction

```

SELECT firm_name, round(avg(audit_effectiveness_score),2) as AvgEffectiveness ,
round(avg(client_satisfaction_score),2) as ClientAvgSatisfaction FROM
big4_financial_risk_compliance GROUP by firm_name ORDER by AvgEffectiveness
DESC

```

Run
SQLite
SQLite.3
SQLite.4
SQLite.5
SQLite.6
SQLite.7

```

1 SELECT firm_name, round(avg(audit_effectiveness_score),2) AS AvgEffectiveness ,
2 round(avg(client_satisfaction_score),2) AS ClientAvgSatisfaction FROM big4_financial_risk_compliance
3 GROUP BY firm_name
4 ORDER BY AvgEffectiveness DESC
5 |

```

Firm_Name	AvgEffectiveness	ClientAvgSatisfaction
KPMG	7.6	7.04
Deloitte	7.52	7.62
Ernst & Young	7.46	7.39
PwC	7.39	7.22

Q10. Impact of AI used and how workload is correlated?

```

SELECT
    ai_used_for_auditing,
    employee_workload, round(avg(audit_effectiveness_score),2) as AvgEffectiveness,
    round(avg(client_satisfaction_score),2) as AvgSatisfaction
from
big4_financial_risk_compliance GROUP by ai_used_for_auditing

```


Run

SQLite

SQLite.3

SQLite.4

SQLite.5

SQLite.6

SQLite.7

```
1 SELECT ai_used_for_auditing, employee_workload, round(avg(audit_effectiveness_score),2) AS AvgEffectiveness,
2 round(avg(client_satisfaction_score),2) AS AvgSatisfaction FROM big4_financial_risk_compliance
3 GROUP BY ai_used_for_auditing
```

AI_Used_for_Auditing	Employee_Workload	AvgEffectiveness	AvgSatisfaction
No	57	7.4	7.32
Yes	58	7.6	7.37

Download

Table

Chart

Raw

Menu

Q1. Count of startups by industry

SELECT industry, count (*) As Startup_count FROM startup_data GROUP by industry
Order by Startup_count DESC



The screenshot shows a SQL query editor interface. At the top, there is a toolbar with a 'Run' button and several tabs labeled 'SQLite.3' through 'SQLite.8'. Below the toolbar, the SQL query is entered in a text area. The query is:
1 SELECT industry, COUNT(*) AS Startup_count FROM startup_data
2 GROUP BY industry
3 ORDER BY Startup_count DESC
Below the query, the results are displayed in a table with two columns: 'Industry' and 'Startup_count'. The results are ordered by 'Startup_count' in descending order.

Industry	Startup_count
EdTech	74
FinTech	71
E-Commerce	70
Gaming	62
AI	62
IoT	61
Cybersecurity	51
HealthTech	49

Q2. Average funding amount by industry?

SELECT industry, round(avg(funding_amount),2) as AvgFundingAmt from startup_data
GROUP by industry ORDER by AvgFundingAmt DESC

Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8				
<pre> 1 SELECT industry, round(avg(funding_amount),2) AS AvgFundingAmt FROM startup_data 2 GROUP BY industry 3 ORDER BY AvgFundingAmt DESC </pre>										
!	Industry	AvgFundingAmt								
	Gaming	173.15								
	E-Commerce	164.63								
	FinTech	164.03								
	Cybersecurity	156.32								
	EdTech	151.52								
	AI	138.34								
	HealthTech	134.93								
	IoT	131.96								

Q3. Total and Average revenue by region?

```

SELECT region , sum(revenue) as TotalRevenue, round(avg(revenue),2) As AvgRevenue
FROM startup_data GROUP by region ORDER by TotalRevenue DESC

```

RunSQLite.3SQLite.4SQLite.5SQLite.6SQLite.7SQLite.8

```
1 SELECT region , sum(revenue) AS TotalRevenue, round(avg(revenue),2) AS AvgRevenue FROM startup_data
2 GROUP BY region
3 ORDER BY TotalRevenue DESC
```

! Region	TotalRevenue	AvgRevenue
Europe	5466.43	53.59
Australia	5088.53	47.12
North America	5030.69	51.86
Asia	4972.65	48.28
South America	4102.57	45.58

Q4. Count of startups by exit status

```
SELECT exit_status,count(*) as Count FROM startup_data GROUP by exit_status
```

Run

SQLite.3

SQLite.4

SQLite.5

SQLite.6

SQLite.7

SQLite.8

```
1 SELECT exit_status,COUNT(*) AS COUNT FROM startup_data
2 GROUP BY exit_status
3
```

! Exit_Status	Count
Acquired	107
IPO	45
Private	348

Q5. Top 5 startups by valuation

```
SELECT * FROM startup_data
```

```
order by valuation DESC
```

limit 5

The screenshot shows a SQL IDE interface. At the top, there is a toolbar with a 'Run' button and several tabs labeled 'SQLite.3' through 'SQLite.8'. Below the toolbar, a query editor contains the following SQL code:

```
1 SELECT * FROM startup_data
2 ORDER BY valuation DESC
3 LIMIT 5
```

Below the query editor, the results are displayed in a table with 12 columns: Star..., Industry, Fundi..., Fundi..., Valuati..., Revenue, Emplo..., Market..., Profita..., Year_..., Region, and Exit_Status. The table contains 5 rows of data, representing the top 5 startups by valuation.

Star...	Industry	Fundi...	Fundi...	Valuati...	Revenue	Emplo...	Market...	Profita...	Year_...	Region	Exit_Status
Startup...	Gaming	2	293.25	4357....	47.48	4001	3.03	0	2022	Asia	Private
Startup...	EdTech	3	288.66	4264.9	9.39	1016	5.22	0	2006	North A...	Private
Startup...	E-Com...	4	285.82	4137....	78.7	2571	8.3	1	2009	Asia	Acquired
Startup...	E-Com...	2	299.81	4125....	77.21	2813	7.56	1	1998	South A...	Acquired
Startup...	E-Com...	2	280.59	4110....	27.85	984	3.52	1	2011	North A...	IPO

Q6. Most profitable industries

SELECT industry, round(avg(profitable),2) as AvgProfitable FROM startup_data

GROUP by industry

order by AvgProfitable DESC

Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8	
-----	----------	----------	----------	----------	----------	----------	--


```

1 SELECT industry, round(avg(profitable),2) AS AvgProfitable FROM startup_data
2 GROUP BY industry
3 ORDER BY AvgProfitable DESC

```


! Industry	AvgProfitable
E-Commerce	0.54
IoT	0.52
AI	0.47
EdTech	0.46
HealthTech	0.45
Gaming	0.34
FinTech	0.34
Cybersecurity	0.31

Q7. Top 10 Market share leaders by industry

```

SELECT industry, market_share_ as MarketSahrePercentage FROM startup_data
ORDER by market_share_ DESC LIMIT

```

Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8	
-----	----------	----------	----------	----------	----------	----------	--

```

1 SELECT industry, market_share_ AS MarketSahrePercentage FROM startup_data
2 ORDER BY market_share_ DESC
3 LIMIT 10

```

! Industry	MarketSahrePercentage
EdTech	10
Gaming	9.96
IoT	9.93
FinTech	9.9
Gaming	9.9
E-Commerce	9.89
AI	9.88
FinTech	9.86
Gaming	9.82

Q8. Funding Trends over time

```
SELECT year_founded , sum(funding_rounds) As Total_Funding_Rounds FROM
startup_data GROUP by year_founded ORDER by year_founded DESC
```


Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8				
<pre> 1 SELECT year_founded , sum(funding_rounds) AS Total_Funding_Rounds FROM startup_data 2 GROUP BY year_founded 3 ORDER BY year_founded DESC </pre>										
! Year_Founded		Total_Funding_Rounds								
2022		21								
2021		65								
2020		37								
2019		41								
2018		52								
2017		45								
2016		51								
2015		54								
2014		36								
2013		30								
2012		58								
2011		46								
2010		28								
2009		49								
2008		44								

Q9. Revenue growth by industry

SELECT industry, round(avg(revenue),2) AS AvgRevenue FROM startup_data

GROUP by industry

Order by AvgRevenue DESC

Run

SQLite.3

SQLite.4

SQLite.5

SQLite.6

SQLite.7

SQLite.8

1 SELECT industry, round(avg(revenue),2) AS AvgRevenue FROM startup_data

2 GROUP BY industry

3 ORDER BY AvgRevenue DESC

Industry	AvgRevenue
Cybersecurity	58.38
HealthTech	51.45
Gaming	51.45
E-Commerce	51.28
FinTech	49.48
EdTech	46.61
AI	46.48
IoT	41.61

Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8	
<pre> 1 SELECT industry, round(avg(funding_amount),2) AS AvgFunding , round(avg(valuation),2) AS AvgValuation FROM startup_data 2 GROUP BY industry 3 ORDER BY AvgFunding DESC </pre>							
Industry	AvgFunding	AvgValuation					
Gaming	173.15	1584.83					
E-Commerce	164.63	1640.42					
FinTech	164.03	1396.91					
Cybersecurity	156.32	1437.18					
EdTech	151.52	1331.93					
AI	138.34	1090.26					
HealthTech	134.93	1240.51					
IoT	131.96	1203.18					

Q11. Profitability across region

SELECT region, round(avg(profitable),2) as AvgProfitability FROM startup_data

GROUP by region

Run

SQLite.3

SQLite.4

SQLite.5

SQLite.6

SQLite.7

SQLite.8

1 SELECT region , round(avg(profitable),2) AS AvgProfitability FROM startup_data

2 GROUP BY region

Region	AvgProfitability
Asia	0.48
Australia	0.48
Europe	0.46
North America	0.38
South America	0.34

Q12. Highly funded startups in specific industries

SELECT industry, funding_amount FROM startup_data WHERE industry in ('AI','Gaming') and funding_amount>250

Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8				
<pre> 1 SELECT industry, funding_amount FROM startup_data 2 WHERE industry IN ('AI','Gaming') AND funding_amount>250 </pre>										
!	Industry	Funding_Amount								
	Gaming	258.39								
	AI	292.38								
	Gaming	251.21								
	Gaming	274.7								
	Gaming	271.06								
	Gaming	298.9								
	AI	269.28								
	Gaming	268.53								
	Gaming	258.11								
	Gaming	255.73								
	AI	272								
	Gaming	288.41								
	AI	284.22								
	Gaming	255.8								
	Gaming	297.05								

\\\\

Q13. Rank startups by revenue within each industry?

SELECT industry, startup_name, revenue, rank () over (partition by industry ORDER by revenue DESC as RankByRevenue

FROM startup_data

Run	SQLite.3	SQLite.4	SQLite.5	SQLite.6	SQLite.7	SQLite.8				
<pre> 1 SELECT industry, startup_name, revenue, rank() over (partition BY industry ORDER BY revenue DESC AS RankByRevenue 2 FROM startup_data 3 4 </pre>										
Industry	Startup_Name	Revenue	RankByRevenue							
AI	Startup_55	99.11	1							
AI	Startup_366	97.08	2							
AI	Startup_299	96.63	3							
AI	Startup_434	90.04	4							
AI	Startup_466	85.2	5							
AI	Startup_144	84.71	6							
AI	Startup_390	83.35	7							
AI	Startup_117	82.8	8							
AI	Startup_303	82.62	9							
AI	Startup_320	82.47	10							
AI	Startup_281	80.76	11							

Q14. Top regions by market share?

SELECT region, sum(market_share_) as TotalMarketShare, rank () over (ORDER by sum(market_share_) DESC) as RankByMarketShare FROM startup_data GROUP by region

