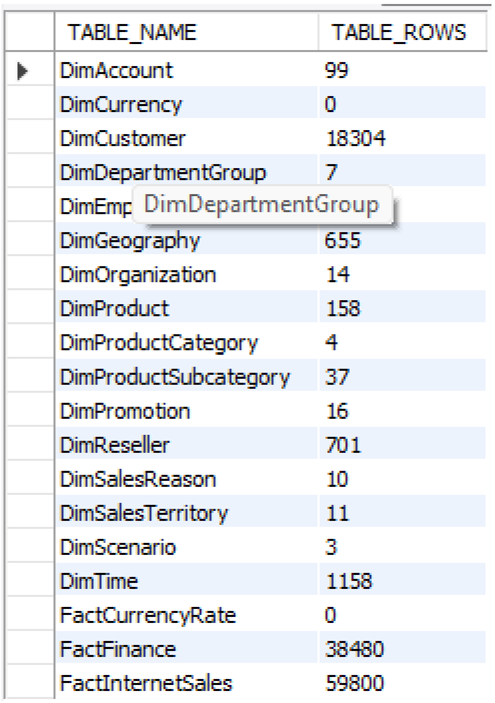
1.1 Use the information_schema to find out how many rows there are in each table in the adventureworks data warehouse. Show the table name and its row count.

Hints:

- use information_schema;
- There is a table within information_schema called TABLES.

```
-- Q1
use information_schema;
Select table_name, table_rows
        from tables
where table_schema like 'aw%';
```



TABLE_NAME	TABLE_ROWS
DimAccount	99
DimCurrency	0
DimCustomer	18304
DimDepartmentGroup	7
DimEmp	
DimGeography	655
DimOrganization	14
DimProduct	158
DimProductCategory	4
DimProductSubcategory	37
DimPromotion	16
DimReseller	701
DimSalesReason	10
DimSalesTerritory	11
DimScenario	3
DimTime	1158
FactCurrencyRate	0
FactFinance	38480
FactInternetSales	59800

1.2 Then run a SELECT COUNT(*) against each table and compare the row count from the SELECT COUNT(*) to the result from the information_schema.

DimAccount	99
DimCurrency	0
DimCustomer	18483
DimDepartmentGroup	7
DimEmployee	296
DimGeography	655
DimOrganization	14

DimProduct	158
DimProductCategory	4
DimProductSubcategory	37
DimPromotion	16
DimReseller	701
DimSalesReason	10
DimSalesTerritory	11
DimScenario	3
DimTime	1158
FactCurrencyRate	0
FactFinance	39410
FactInternetSales	60398

1.3 Do some research to find out why MySQL might return different values in such a scenario. Document your findings. In other words, explain WHY is the row count from the information_schema might sometimes be different from the SELECT COUNT(*).

Based on my research, information_schema usually estimate the count of rows, so the value we saw with information_schema is an estimated value. On the other hands, information_schema may only count NON_Null value while SELECT COUNT(*) doesn't. The row count of information_schema may be calculated based on the statistical sampling techniques to estimate the number.

1.4 Why is the select against the information_schema more efficient (although perhaps less accurate) than the SELECT COUNT(*)?

Hints:

• This AW schema was created using the "innodb" database engine.

With such a database engine, it can provide an estimated row count based on statistical sampling without scanning the entire table.

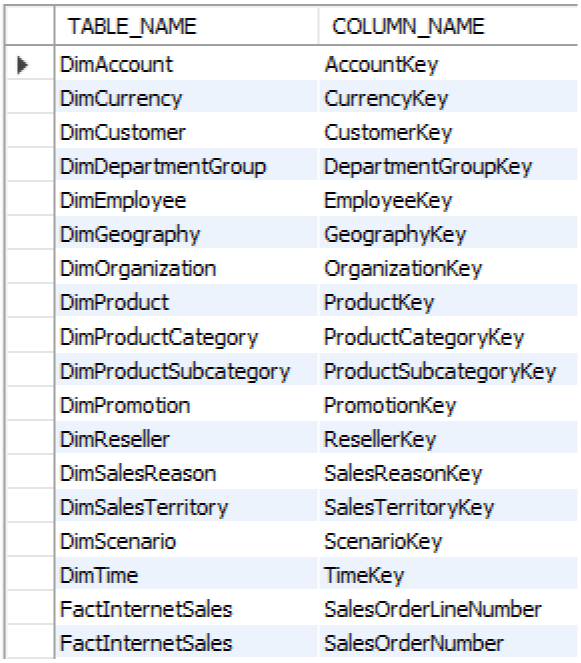
SELECT COUNT(*) locks and scans the whole table, which takes more time to compute the exact number of rows.

And information_schema is doing an estimation. It is less-accuracy but more time-efficiency.

2. Use the information_schema to list out each table in the adventureworks data warehouse and its primary key.

Hints:

- There is a table within information_schema called COLUMNS.
- Look for a column called column_key



	TABLE_NAME	COLUMN_NAME
▶	DimAccount	AccountKey
	DimCurrency	CurrencyKey
	DimCustomer	CustomerKey
	DimDepartmentGroup	DepartmentGroupKey
	DimEmployee	EmployeeKey
	DimGeography	GeographyKey
	DimOrganization	OrganizationKey
	DimProduct	ProductKey
	DimProductCategory	ProductCategoryKey
	DimProductSubcategory	ProductSubcategoryKey
	DimPromotion	PromotionKey
	DimReseller	ResellerKey
	DimSalesReason	SalesReasonKey
	DimSalesTerritory	SalesTerritoryKey
	DimScenario	ScenarioKey
	DimTime	TimeKey
	FactInternetSales	SalesOrderLineNumber
	FactInternetSales	SalesOrderNumber

3. What standard table naming convention did the AdventureWorksDW database designers use to differentiate dimension tables from fact tables in this star schema data warehouse?

All the dimension tables have been prefixed with "Dim"

All the fact tables have been prefixed with "Fact"

In this star schema data warehouse, the prefix differentiates dimension tables from fact tables.

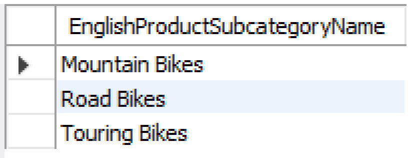
4. What is the purpose of the recursive relationship on DimEmployee?

The purpose of the recursive relationship is to represent the hierarchical relationship between employees. Employee can report to their boss, who is another employee shown in the table. In other words, the recursive relationship also reflects the parent-child relationship between employees.

5. What are the three types of models of bikes sold by AdventureWorks? Provide your SQL query, and your answer set along with your answer to the question. (HINT: There are many models, but all those models fall into just three major types of bikes.)

Mountain bikes, Road Bikes, Touring Bikes

```
SELECT DISTINCT EnglishProductSubcategoryName
          FROM DimProductSubcategory
INNER JOIN DimProductCategory
          ON DimProductSubcategory.ProductCategoryKey = DimProductCategory.ProductCategoryKey
WHERE DimProductCategory.EnglishProductCategoryName = 'Bikes';
```

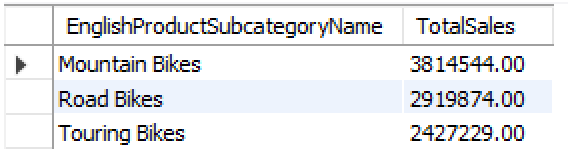


	EnglishProductSubcategoryName
▶	Mountain Bikes
	Road Bikes
	Touring Bikes

6. Of these three, which type of bike model had the highest sales (in dollar volume) in 2004? Provide your SQL query, and your answer set along with your answer to the question.

Mountain Bikes has the highest sales.

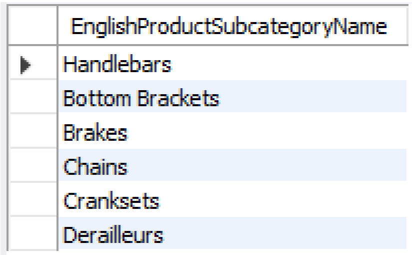
```
SELECT DISTINCT EnglishProductSubcategoryName,  SUM(FactInternetSales.SalesAmount) AS
'TotalSales'
          FROM DimProductSubcategory
INNER JOIN DimProduct
          ON DimProductSubcategory. ProductSubcategoryKey = DimProduct.ProductSubcategoryKey
INNER JOIN FactInternetSales
          ON DimProduct.ProductKey = FactInternetSales.ProductKey
INNER JOIN DimTime
          ON FactInternetSales.OrderDateKey = DimTime.TimeKey
WHERE DimProductSubcategory.ProductCategoryKey = '1'
          AND
                    DimTime.CalendarYear = '2004'
GROUP BY EnglishProductSubcategoryName;
```



	EnglishProductSubcategoryName	TotalSales
▶	Mountain Bikes	3814544.00
	Road Bikes	2919874.00
	Touring Bikes	2427229.00

7. List six of the other non-bike products sold by AdventureWorks. (Pick any six.) Provide your SQL query, and your answer set along with your answer to the question.

```
SELECT EnglishProductSubcategoryName
        FROM DimProductSubcategory
WHERE ProductCategoryKey != 1
LIMIT 6;
```



	EnglishProductSubcategoryName
▶	Handlebars
	Bottom Brackets
	Brakes
	Chains
	Cranksets
	Derailleurs

8. Compare and rank the total counts of the bikes sold by AdventureWorks for each of the years 2001 – 2004 by color. What was the most popular color of bikes sold in each of these 4 years? Provide your SQL query, and your answer set along with your answer to the question. You can assume that one row in the fact table equals one sale.

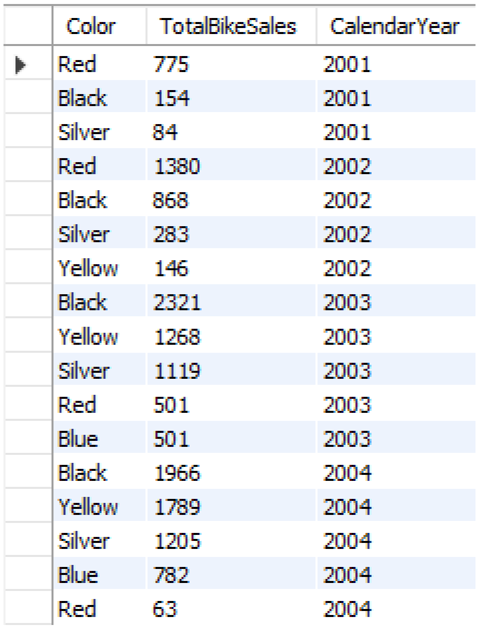
Red bike is the best sold of 2001

Red bike is the best sold of 2002

Black bike is the best sold of 2001

Black bike is the best sold of 2001

```
SELECT DISTINCT DimProduct.Color, COUNT(FactInternetSales.OrderQuantity) AS 'TotalBikeSales',
DimTime.CalendarYear

          FROM DimProductSubcategory

INNER JOIN DimProduct

          ON DimProductSubcategory. ProductSubcategoryKey = DimProduct.ProductSubcategoryKey

INNER JOIN FactInternetSales

          ON DimProduct.ProductKey = FactInternetSales.ProductKey

INNER JOIN DimTime

          ON FactInternetSales.OrderDateKey = DimTime.TimeKey

WHERE DimProductSubcategory.ProductCategoryKey = '1'

          AND DimTime.CalendarYear BETWEEN 2001 AND 2004


GROUP BY DimProduct.Color,DimTime.CalendarYear

ORDER BY DimTime.CalendarYear,

                    COUNT(FactInternetSales.OrderQuantity) DESC;
```



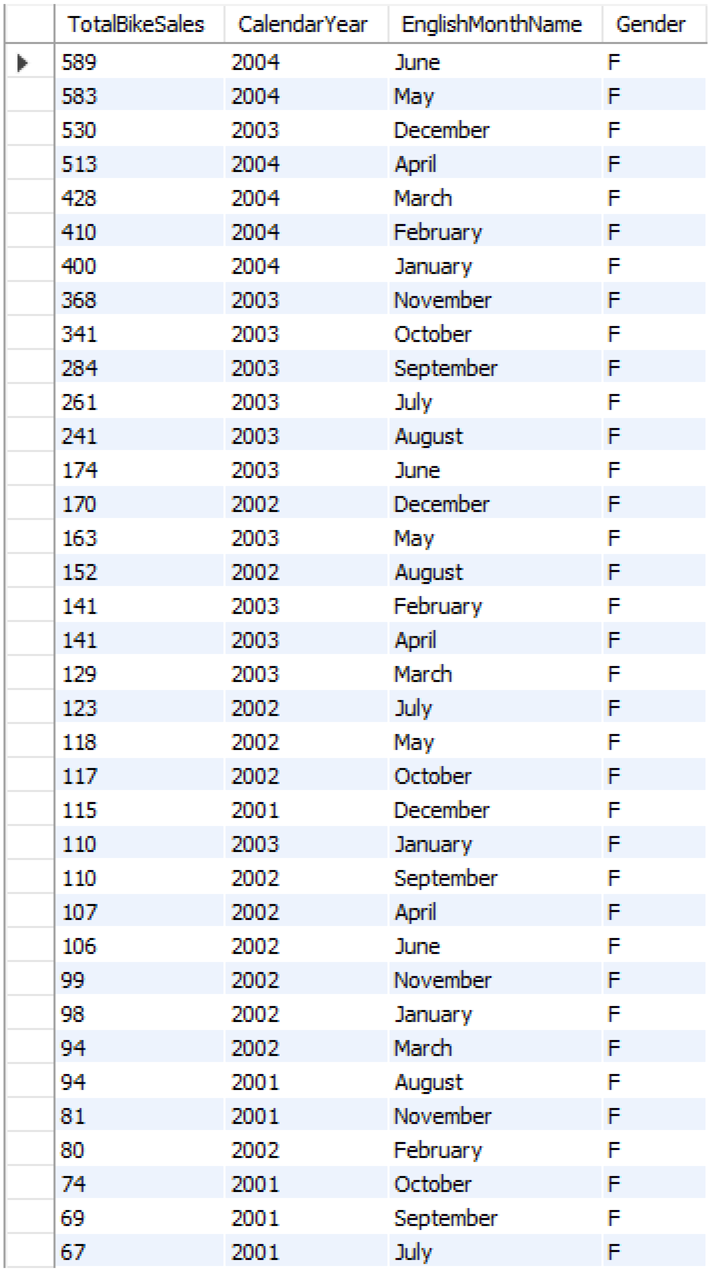
Color	TotalBikeSales	CalendarYear
Red	775	2001
Black	154	2001
Silver	84	2001
Red	1380	2002
Black	868	2002
Silver	283	2002
Yellow	146	2002
Black	2321	2003
Yellow	1268	2003
Silver	1119	2003
Red	501	2003
Blue	501	2003
Black	1966	2004
Yellow	1789	2004
Silver	1205	2004
Blue	782	2004
Red	63	2004

HINT: Since the fact table contains sales for all kinds of products, you should include only fact rows where the sale is for a bike. One easy way to do this is a WHERE clause selecting only rows where EnglishProductSubcategoryName contains the string “bikes”. There are other ways to determine this as well.

9. List and compare the total sales quantities (number of bikes, NOT dollars) of bikes sold (all model types) by customer gender by year and month. In which year and month were bike sales to females the highest? Provide your SQL query, and your answer set along with your answer to the question.

June, 2004 has the highest sales.

```
SELECT DISTINCT  COUNT(*) AS 'TotalBikeSales', DimTime.CalendarYear,
DimTime.EnglishMonthName, DimCustomer.Gender
            FROM DimProductSubcategory
INNER JOIN DimProduct
            ON DimProductSubcategory. ProductSubcategoryKey = DimProduct.ProductSubcategoryKey
INNER JOIN FactInternetSales
            ON DimProduct.ProductKey = FactInternetSales.ProductKey
INNER JOIN DimCustomer
            ON DimCustomer.CustomerKey = FactInternetSales.CustomerKey
INNER JOIN DimTime
            ON FactInternetSales.OrderDateKey = DimTime.TimeKey
WHERE DimCustomer.Gender = 'F'
            AND DimProductSubcategory.ProductCategoryKey = '1'
GROUP BY DimCustomer.Gender, DimTime.CalendarYear, DimTime.EnglishMonthName
ORDER BY COUNT(*) DESC;
```

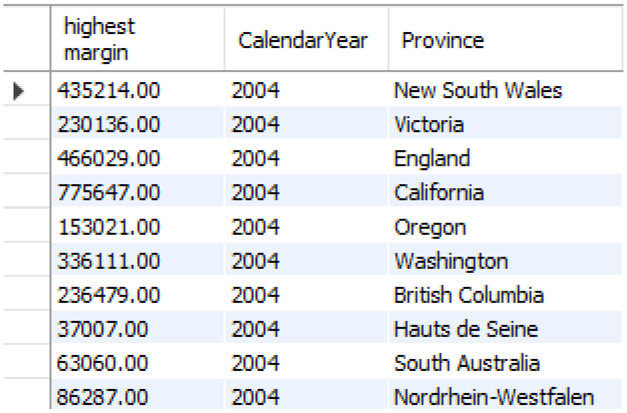


TotalBikeSales	CalendarYear	EnglishMonthName	Gender
589	2004	June	F
583	2004	May	F
530	2003	December	F
513	2004	April	F
428	2004	March	F
410	2004	February	F
400	2004	January	F
368	2003	November	F
341	2003	October	F
284	2003	September	F
261	2003	July	F
241	2003	August	F
174	2003	June	F
170	2002	December	F
163	2003	May	F
152	2002	August	F
141	2003	February	F
141	2003	April	F
129	2003	March	F
123	2002	July	F
118	2002	May	F
117	2002	October	F
115	2001	December	F
110	2003	January	F
110	2002	September	F
107	2002	April	F
106	2002	June	F
99	2002	November	F
98	2002	January	F
94	2002	March	F
94	2001	August	F
81	2001	November	F
80	2002	February	F
74	2001	October	F
69	2001	September	F
67	2001	July	F

10. For the year 2004, which State/Province yielded the highest margin for AdventureWorks? (HINT: use the customer's State/Province.) Provide your SQL query, and your answer set along with your answer to the question. Margin is the difference between what Adventureworks PAID for a bike and what they SOLD it for.

New South Wales yield the highest margin for AdventureWorks

```
SELECT DISTINCT  SUM((FactInternetSales.UnitPrice - FactInternetSales.ProductStandardCost) ) AS
'highest margin', DimTime.CalendarYear,  DimGeography.StateProvinceName AS Province

            FROM DimProductSubcategory

INNER JOIN DimProduct

            ON DimProductSubcategory. ProductSubcategoryKey = DimProduct.ProductSubcategoryKey

INNER JOIN FactInternetSales

            ON DimProduct.ProductKey = FactInternetSales.ProductKey

INNER JOIN DimCustomer

            ON DimCustomer.CustomerKey = FactInternetSales.CustomerKey

INNER JOIN DimTime

            ON FactInternetSales.OrderDateKey = DimTime.TimeKey

INNER JOIN DimGeography

            On DimGeography.GeographyKey = DimCustomer.GeographyKey

WHERE DimTime.CalendarYear = '2004'

            AND DimProductSubcategory.ProductCategoryKey = '1'

GROUP BY DimGeography.StateProvinceName

ORDER BY 'highest margin'  DESC

LIMIT 10;
```



highest margin	CalendarYear	Province
435214.00	2004	New South Wales
230136.00	2004	Victoria
466029.00	2004	England
775647.00	2004	California
153021.00	2004	Oregon
336111.00	2004	Washington
236479.00	2004	British Columbia
37007.00	2004	Hauts de Seine
63060.00	2004	South Australia
86287.00	2004	Nordrhein-Westfalen