

DATABASE PROPOSITIONS AND SOLUTIONS

Problem statements for exploring and creating insights from the newly acquired databases: AdventureWorks2014, AdventureWorksDW2014, and Northwinds2019TSQLV5.

CSCI 331
Group 12-5
Ibrahim Suhail

DATABASE PROPOSITIONS AND SOLUTIONS

Contents

Proposition 1	9
Diagram of Database	9
Problem Statement	9
Database	9
Attributes	11
Problem-solving Query	11
Sample Relational Output	11
Sample JSON Output	11
Proposition 2	12
Diagram of Database	12
Problem Statement	13
Database	13
Diagrams of Tables	14
Key View	14
Standard View	15
Attributes	15
Problem-solving Query	15
Sample Relational Output	16
Sample JSON Output	16
Proposition 3	17
Diagram of Database	17
Problem Statement	18
Database	18
Diagrams of Tables	18
Key View	18
Attributes	19

DATABASE PROPOSITIONS AND SOLUTIONS

Problem-solving Query	19
Sample Relational Output	20
Sample JSON Output	20
Proposition 4	20
Diagram of Database	21
Database	21
Diagrams of Tables	22
Key View	22
Standard View	23
Attributes	23
Problem-solving Query	23
Sample Relational Output	24
Sample JSON Output	24
Proposition 5	25
Diagram of Database	26
Problem Statement	26
Database	26
Diagrams of Tables	27
Key View	27
Standard View	27
Attributes	28
Problem-solving Query	28
Sample Relational Output	28
Sample JSON Output	28
Proposition 6	30
Diagram of Database	30
Problem Statement	30

DATABASE PROPOSITIONS AND SOLUTIONS

Database	30
Diagrams of Tables	30
Key View	31
Standard View	31
Attributes	31
Problem-solving Query	32
Sample Relational Output	32
Sample JSON Output	32
Proposition 7	32
Diagram of Database	34
Problem Statement	34
Database	34
Diagrams of Tables	34
Key View	34
Standard View	35
Attributes	35
Order by	36
Problem-solving Query	36
Sample Relational Output	36
Sample JSON Output	37
Proposition 8	37
Diagram of Database	37
Problem Statement	38
Database	38
Diagrams of Tables	38
Key View	38
Standard View	39

DATABASE PROPOSITIONS AND SOLUTIONS

Attributes	39
Problem-solving Query	40
Sample Relational Output	41
Sample JSON Output	41
Proposition 9	41
Diagram of Database	42
Problem Statement	43
Database	43
Diagrams of Tables	43
Key Views	43
Standard View	44
Attributes	44
Problem-solving Query	44
Sample Relational Output	45
Sample JSON Output	46
Proposition 10	46
Diagram of Database	46
Problem Statement	47
Database	47
Diagrams of Tables	47
Key View	47
Standard View	48
Attributes	48
Problem-solving Query	49
Sample JSON Output	50
Proposition 11	50
Diagram of Database	52

DATABASE PROPOSITIONS AND SOLUTIONS

Problem Statement	52
Database	52
Diagrams of Tables	52
Key View	52
Standard View	53
Attributes	53
Problem-solving Query	54
Sample Relational Output	55
Sample JSON Output	55
Proposition 12	55
Diagram of Database	57
Problem Statement	57
Database	57
Diagrams of Tables	57
Key View	57
Standard View	58
Attributes	58
Problem-solving Query	59
Sample Relational Output	60
Sample JSON Output	60
Proposition 13	60
Diagram of Database	62
Problem Statement	62
Database	62
Attributes	62
Problem-solving Query	63
Sample Relational Output	63

DATABASE PROPOSITIONS AND SOLUTIONS

Sample JSON Output	64
Proposition 14	64
Diagram of Database	65
Problem Statement	66
Database	66
Diagrams of Tables	66
Key View	66
Standard View	68
Attributes	68
Problem-solving Query	68
Sample Relational Output	69
Sample JSON Output	69
Proposition 15	69
Diagram of Database	71
Problem Statement	71
Database	71
Diagrams of Tables	71
Key View	71
Standard View	72
Attributes	72
Problem-solving Query	72
Sample Relational Output	73
Sample JSON Output	74
Proposition 16	74
Diagram of Database	76
Problem Statement	76
Database	76

DATABASE PROPOSITIONS AND SOLUTIONS

Attributes	76
Order by	76
Problem-solving Query	76
Sample Relational Output	76
Sample JSON Output	77
Proposition 17	77
Diagram of Database	78
Problem Statement	79
Database	79
Attributes	79
Problem-solving Query	79
Sample Relational Output	79
Sample JSON Output	80
Proposition 18	80
Diagram of Database	81
Problem Statement	82
Database	82
Attributes	82
Problem-solving Query	82
Sample Relational Output	82
Sample JSON Output	83
Proposition 19	83
Diagram of Database	85
Problem Statement	85
Database	85
Attributes	85
Problem-solving Query	85

DATABASE PROPOSITIONS AND SOLUTIONS

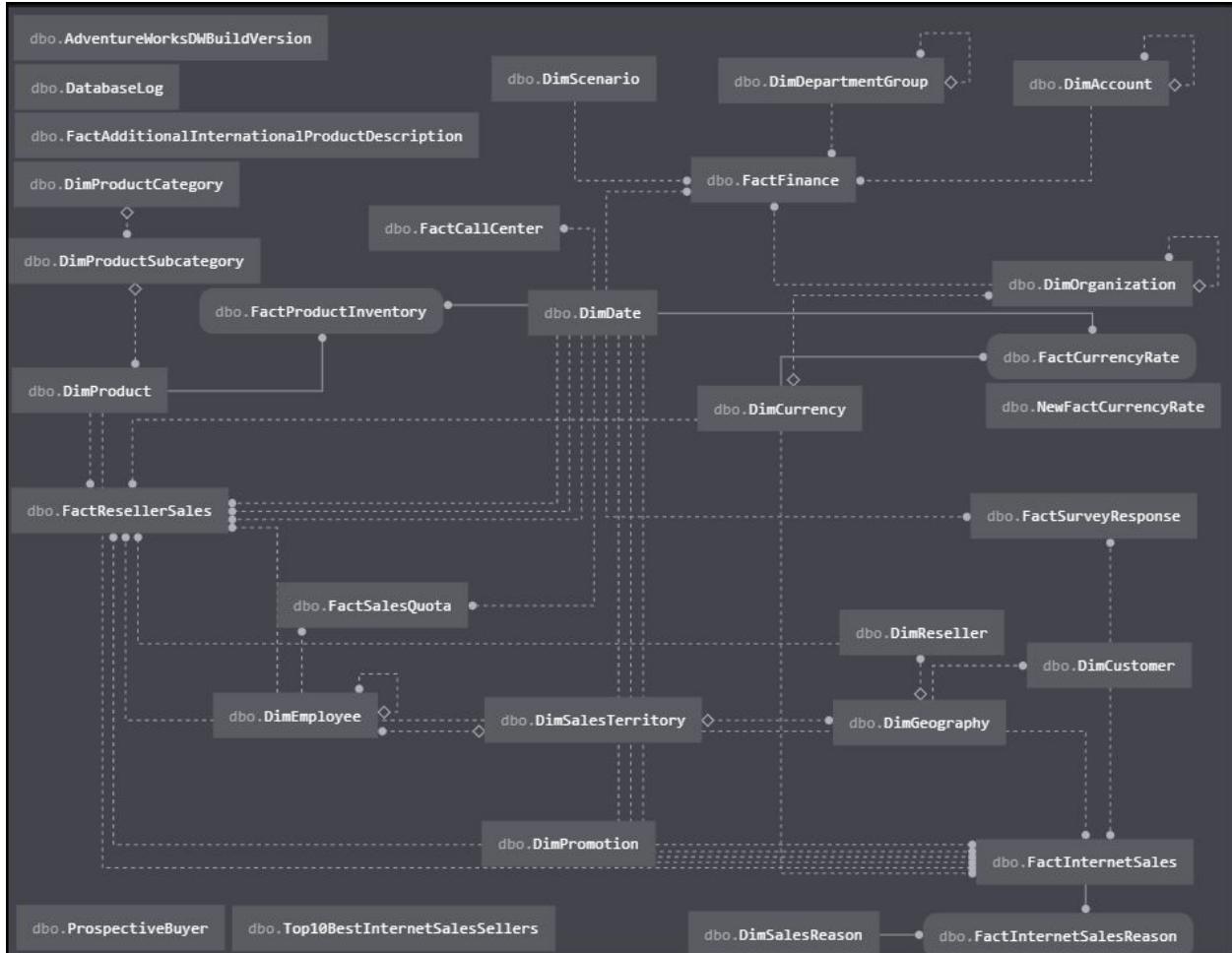
Sample Relational Output	86
Sample JSON Output	86
Proposition 20	86
Diagram of Database	88
Problem Statement	88
Database	88
Diagrams of Tables	88
Key View	88
Standard View	89
Attributes	89
Problem-solving Query	89
Sample Relational Output	90
Sample JSON Output	90

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 1

Diagram of Database

dbo sub-system in AdventureWorksDW2014



Problem Statement

For each year find the average employee base salary rate

Database

AdventureWorksDW2014 database

Prepared by: Ibrahim Suhail

Date Prepared: 3/21/2020

C:\Users\Ibrahim Suhail\Desktop\Spring 2020\CS331\Project 1\G12-5 - Ibrahim Suhail.docx

Page 9 of 88

DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View

dbo.FactInternetSales		
SalesOrderNumber	nvarchar(20)	PK
SalesOrderLineNumber	tinyint	PK
ProductKey	int	FK
OrderDateKey	int	FK
DueDateKey	int	FK
ShipDateKey	int	FK
CustomerKey	int	FK
PromotionKey	int	FK
CurrencyKey	int	FK
SalesTerritoryKey	int	FK
EmployeeKey	int	FK
dbo.DimEmployee		
EmployeeKey	int IDENTITY	PK
SalesTerritoryKey	int	NULL FK

Standard View

dbo.FactInternetSales			dbo.DimEmployee		
SalesOrderNumber	nvarchar(20)	PK	EmployeeKey	int IDENTITY	PK
SalesOrderLineNumber	tinyint	PK	EmployeeNationalIDAlternateKey	nvarchar(15)	NULL
ProductKey	int	FK	ParentEmployeeNationalIDAlternateKey	nvarchar(15)	NULL
OrderDateKey	int	FK	SalesTerritoryKey	int	NULL FK
DueDateKey	int	FK	FirstName	nvarchar(50)	
ShipDateKey	int	FK	LastName	nvarchar(50)	
CustomerKey	int	FK	MiddleName	nvarchar(50)	NULL
PromotionKey	int	FK	NameStyle	bit	
CurrencyKey	int	FK	Title	nvarchar(50)	NULL
SalesTerritoryKey	int	FK	HireDate	date	NULL
RevisionNumber	tinyint		BirthDate	date	NULL
OrderQuantity	smallint		LoginID	nvarchar(256)	NULL
UnitPrice	money		EmailAddress	nvarchar(50)	NULL
ExtendedAmount	money		Phone	nvarchar(25)	NULL
UnitPriceDiscountPct	float		MaritalStatus	nchar(1)	NULL
DiscountAmount	float		EmergencyContactName	nvarchar(50)	NULL
ProductStandardCost	money		EmergencyContactPhone	nvarchar(25)	NULL
TotalProductCost	money		SalariedFlag	bit	NULL
SalesAmount	money		Gender	nchar(1)	NULL
TaxAmt	money		PayFrequency	tinyint	NULL
Freight	money		BaseRate	money	NULL
CarrierTrackingNumber	nvarchar(25)	NULL	VacationHours	smallint	NULL
CustomerPONumber	nvarchar(25)	NULL	SickLeaveHours	smallint	NULL
OrderDate	datetime	NULL	CurrentFlag	bit	
DueDate	datetime	NULL	SalesPersonFlag	bit	
ShipDate	datetime	NULL	DepartmentName	nvarchar(50)	NULL
EmployeeKey	int	FK	StartDate	date	NULL
			EndDate	date	NULL
			Status	nvarchar(50)	NULL
			EmployeePhoto	varbinary(max)	NULL

DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
FactInternetSales	OrderYear
DimEmployee	AVERAGEEMPLOYEEBASERATE

Problem-solving Query

```
USE AdventureWorksDW2014
GO
SELECT year(OrderDate) AS [OrderYear]
      ,avg(e.baserate) AS [AVERAGEEMPLOYEEBASERATE]
FROM FactInternetSales
JOIN DimEmployee AS e ON year(OrderDate) = year(HireDate)
GROUP BY year(OrderDate)
```

Sample Relational Output

Results	Messages
1 2010 26.4065	
2 2011 31.4182	
3 2012 29.3329	

Ln 74, Col 27 Spaces: 4 UTF-8 LF SQL MSSQL 3 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Employee')
```

DATABASE PROPOSITIONS AND SOLUTIONS

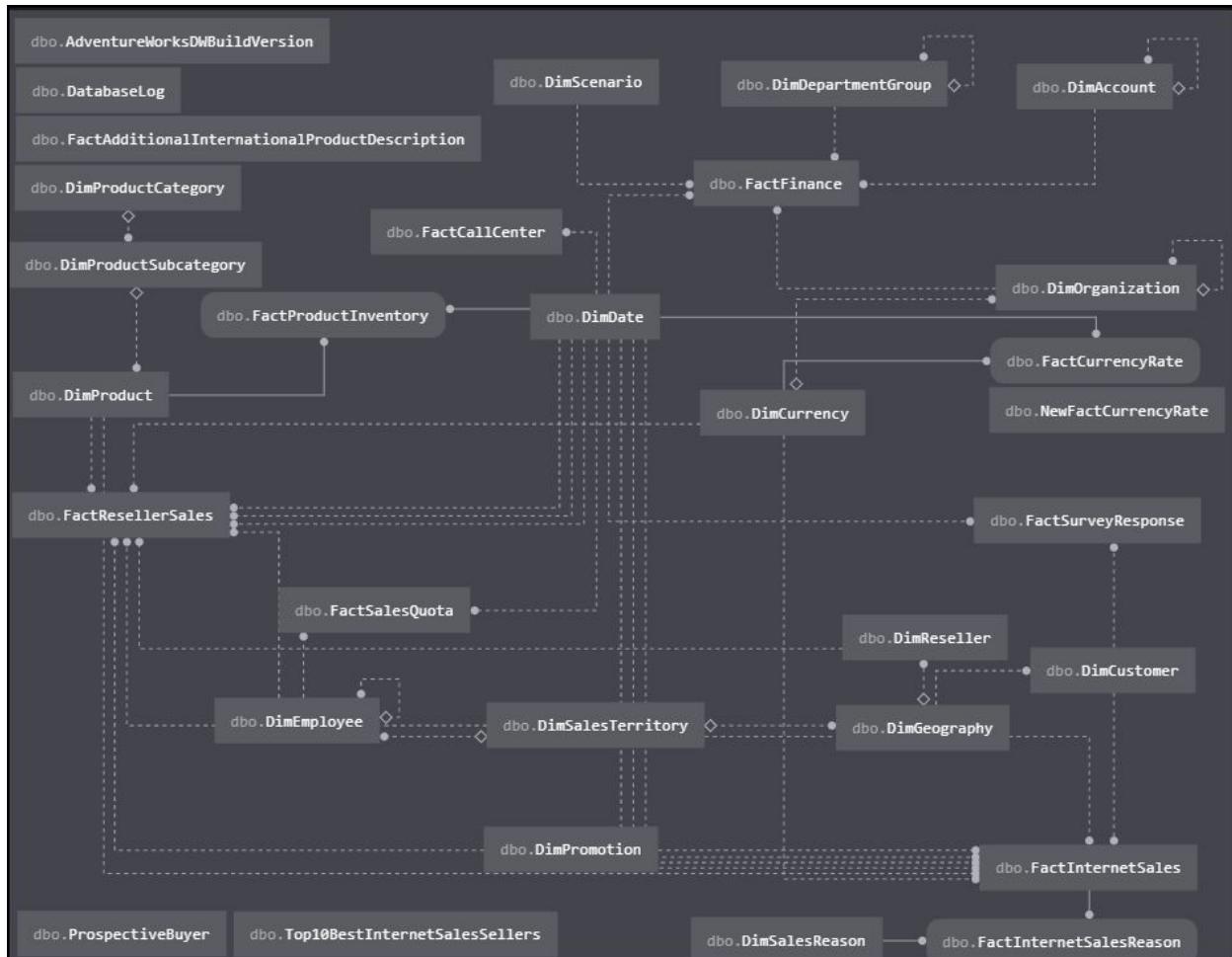
```
1  [
2      "Employee": [
3          {
4              "OrderYear": 2010,
5              "AVERAGEEMPLOYEEBASERATE": 26.4065
6          },
7          {
8              "OrderYear": 2011,
9              "AVERAGEEMPLOYEEBASERATE": 31.4182
10         },
11         {
12             "OrderYear": 2012,
13             "AVERAGEEMPLOYEEBASERATE": 29.3329
14         }
15     ]
16 ]
```

Proposition 2

Diagram of Database

dbo sub-system in AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS



Problem Statement

Find the total sales amount per state

Database

AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



DATABASE PROPOSITIONS AND SOLUTIONS

Standard View

dbo.FactInternetSales		
SalesOrderNumber	nvarchar(20)	PK
SalesOrderLineNumber	tinyint	PK
GeographyKey	int	PK FK
ProductKey	int	FK
OrderDateKey	int	FK
DueDateKey	int	FK
ShipDateKey	int	FK
CustomerKey	int	FK
PromotionKey	int	FK
CurrencyKey	int	FK
SalesTerritoryKey	int	FK
RevisionNumber	tinyint	
OrderQuantity	smallint	
UnitPrice	money	
ExtendedAmount	money	
UnitPriceDiscountPct	float	
DiscountAmount	float	
ProductStandardCost	money	
TotalProductCost	money	
SalesAmount	money	
TaxAmt	money	
Freight	money	
CarrierTrackingNumber	nvarchar(25)	NULL
CustomerPONumber	nvarchar(25)	NULL
OrderDate	datetime	NULL
DueDate	datetime	NULL
ShipDate	datetime	NULL
EmployeeKey	int	FK

dbo.DimGeography		
GeographyKey	int	IDENTITY PK
City	nvarchar(30)	NULL
StateProvinceCode	nvarchar(3)	NULL
StateProvinceName	nvarchar(50)	NULL
EnglishCountryRegionName_1	nvarchar(50)	NULL
CountryRegionCode	nvarchar(3)	NULL
EnglishCountryRegionName	nvarchar(50)	NULL
SpanishCountryRegionName	nvarchar(50)	NULL
FrenchCountryRegionName	nvarchar(50)	NULL
PostalCode	nvarchar(15)	NULL
SalesTerritoryKey	int	NULL FK
IpAddressLocator	nvarchar(15)	NULL

Attributes

Table Name	Column Name
dimgeography	STATE
FactInternetSales	TOTAL SALES

Problem-solving Query

```
USE AdventureWorksDW2014
```

```
GO
```

```

SELECT a.StateProvinceName AS [STATE]
      ,sum(b.SalesAmount) AS [TOTAL SALES]
FROM dbo.dimgeography AS a
JOIN FactInternetSales AS b ON a.SalesTerritoryKey = b.SalesTerritoryKey
GROUP BY a.StateProvinceName
  
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	STATE	TOTAL SALES
1	Alabama	61194.2480
2	Alberta	3955689.7242
3	Arizona	40027055.6854
4	Bayern	17365874.0292
5	Brandenburg	5788624.6764
6	British Columbia	57357501.0009
7	Brunswick	1977844.8621
8	California	537506176.3468
9	Charente-Maritime	2644017.7143
10	Colorado	21005.8072

Ln 79, Col 1 (241 selected) Spaces: 4 UTF-8 LF SQL MSSQL 71 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('State')
```

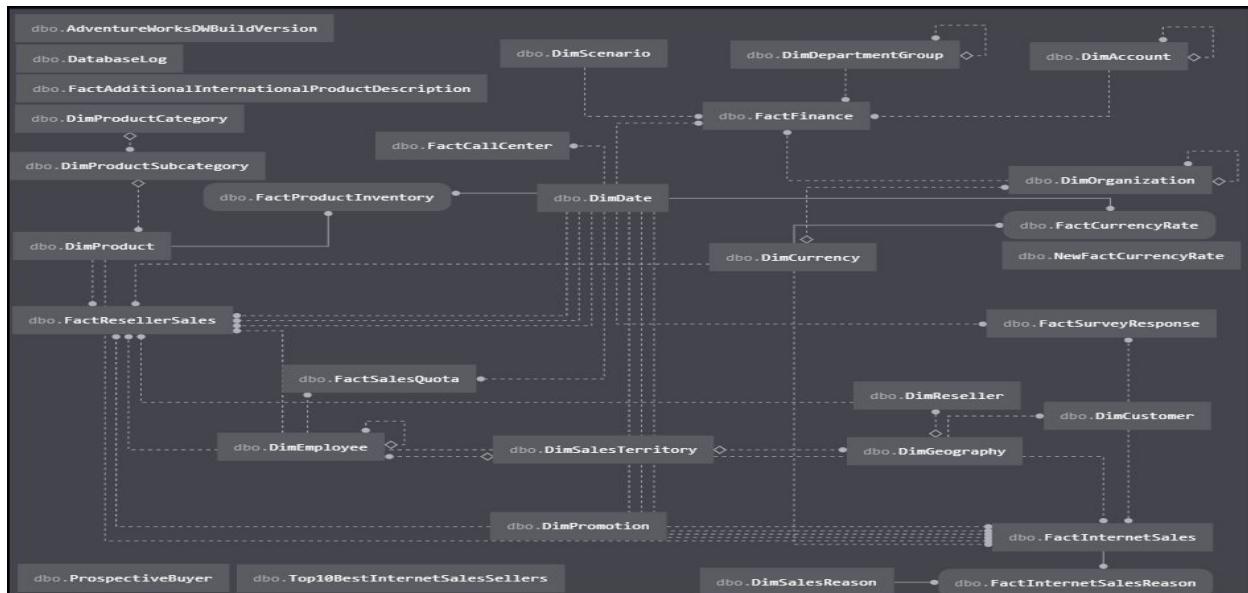
DATABASE PROPOSITIONS AND SOLUTIONS

```
1  [
2    "State": [
3      {
4        "STATE": "Alabama",
5        "TOTAL SALES": 61194.2480
6      },
7      {
8        "STATE": "Alberta",
9        "TOTAL SALES": 3955689.7242
10     },
11     {
12       "STATE": "Arizona",
13       "TOTAL SALES": 40027055.6854
14     },
15     {
16       "STATE": "Bayern",
17       "TOTAL SALES": 17365874.0292
18     },
19     {
20       "STATE": "Brandenburg",
21       "TOTAL SALES": 5788624.6764
22     },
23     {
24       "STATE": "British Columbia",
25       "TOTAL SALES": 57357501.0009
26     }
],
```

Proposition 3

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Prepared by: Ibrahim Suhail

Date Prepared: 3/21/2020

C:\Users\Ibrahim Suhail\Desktop\Spring 2020\CS331\Project 1\G12-5 - Ibrahim Suhail.docx

Page 17 of 88

DATABASE PROPOSITIONS AND SOLUTIONS

Problem Statement

Find the average freight cost per year

Database

AdventureWorksDW2014

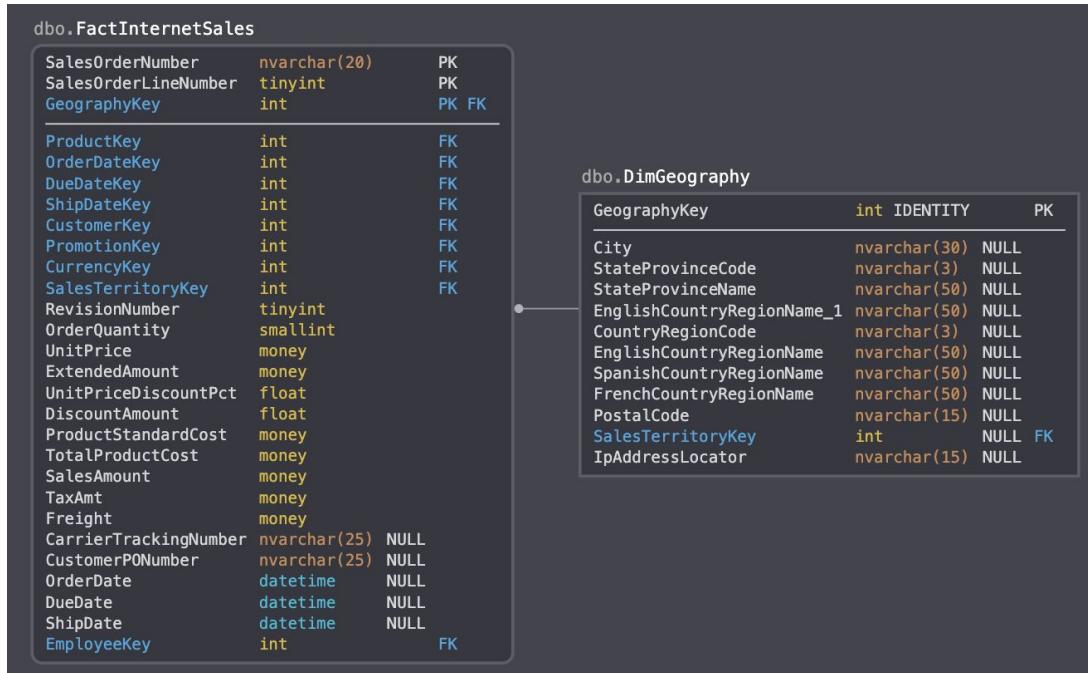
Diagrams of Tables

Key View



DATABASE PROPOSITIONS AND SOLUTIONS

Standard View



Attributes

Table Name	Column Name
FactInternetSales	AVG FREIGHT COST YEAR

Problem-solving Query

```
USE AdventureWorksDW2014
GO
```

```
SELECT AVG(Freight) AS [AVG FREIGHT COST]
    ,YEAR(OrderDate) AS [YEAR]
FROM dbo.dimgeography AS a
JOIN FactInternetSales AS b ON a.SalesTerritoryKey = b.SalesTerritoryKey
GROUP BY YEAR(OrderDate)
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	Avg Freight Cost	Year
1	75.4269	2010
2	7.4668	2013
3	0.5778	2014
4	79.7248	2011
5	42.0475	2012

Ln 86, Col 4 (9 selected) Spaces: 4 UTF-8 LF SQL MSSQL 5 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Freight')
```

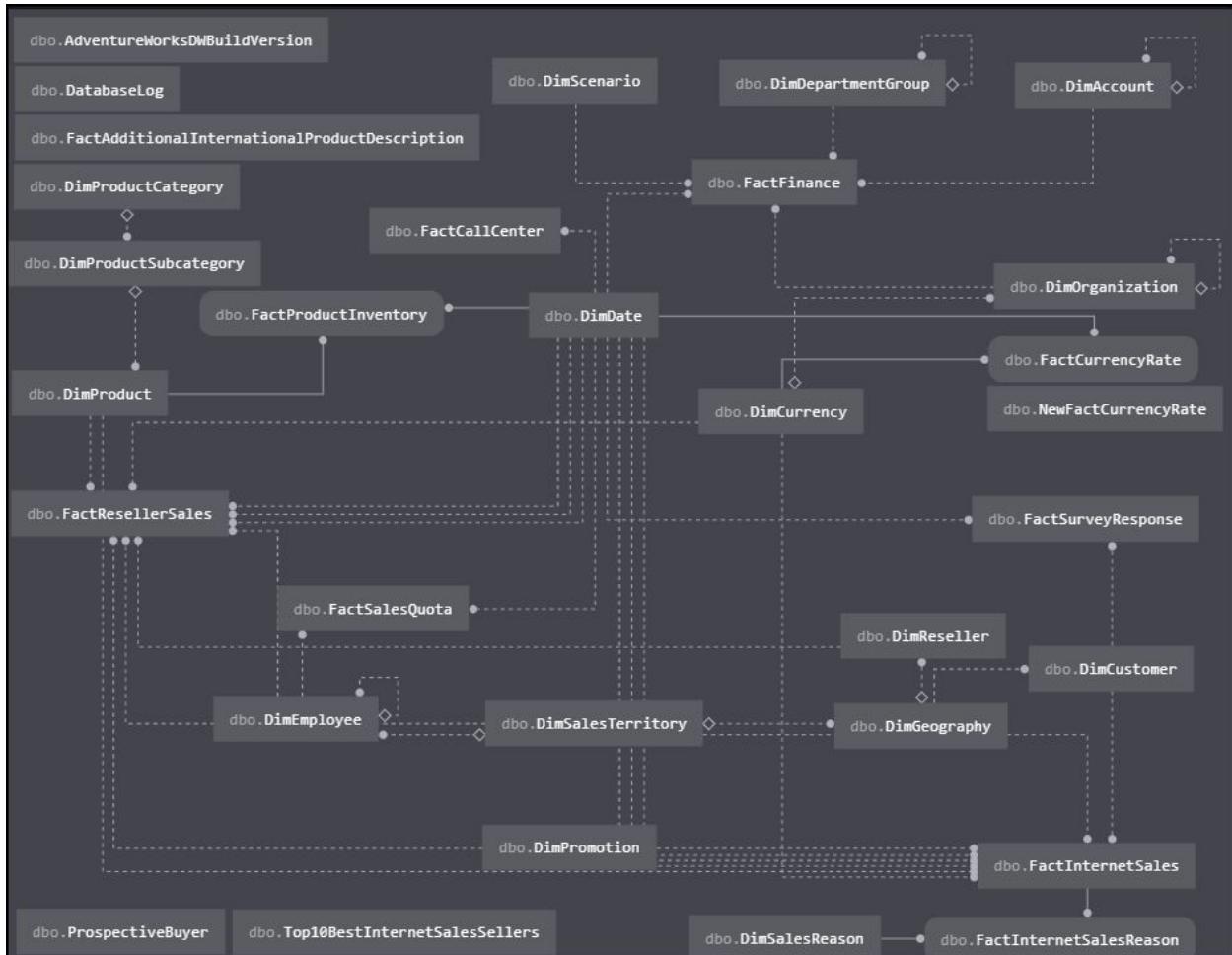
```
1  [
2      "Freight": [
3          {
4              "AVG FEIGHT COST": 75.4269,
5              "YEAR": 2010
6          },
7          {
8              "AVG FEIGHT COST": 7.4668,
9              "YEAR": 2013
10         },
11         {
12             "AVG FEIGHT COST": 0.5778,
13             "YEAR": 2014
14         },
15         {
16             "AVG FEIGHT COST": 79.7248,
17             "YEAR": 2011
18         },
19         {
20             "AVG FEIGHT COST": 42.0475,
21             "YEAR": 2012
22         }
23     ]
24 ]
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 4

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Problem Statement

Find the average number of cars owned per customer in 67 cities.

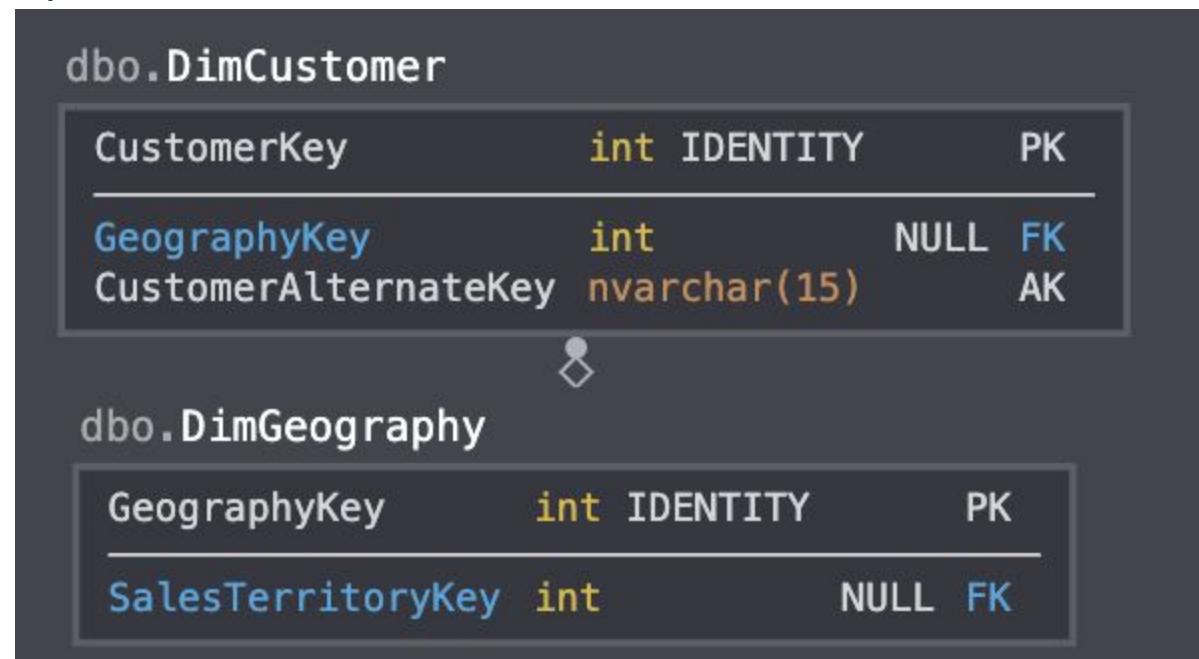
Database

AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS

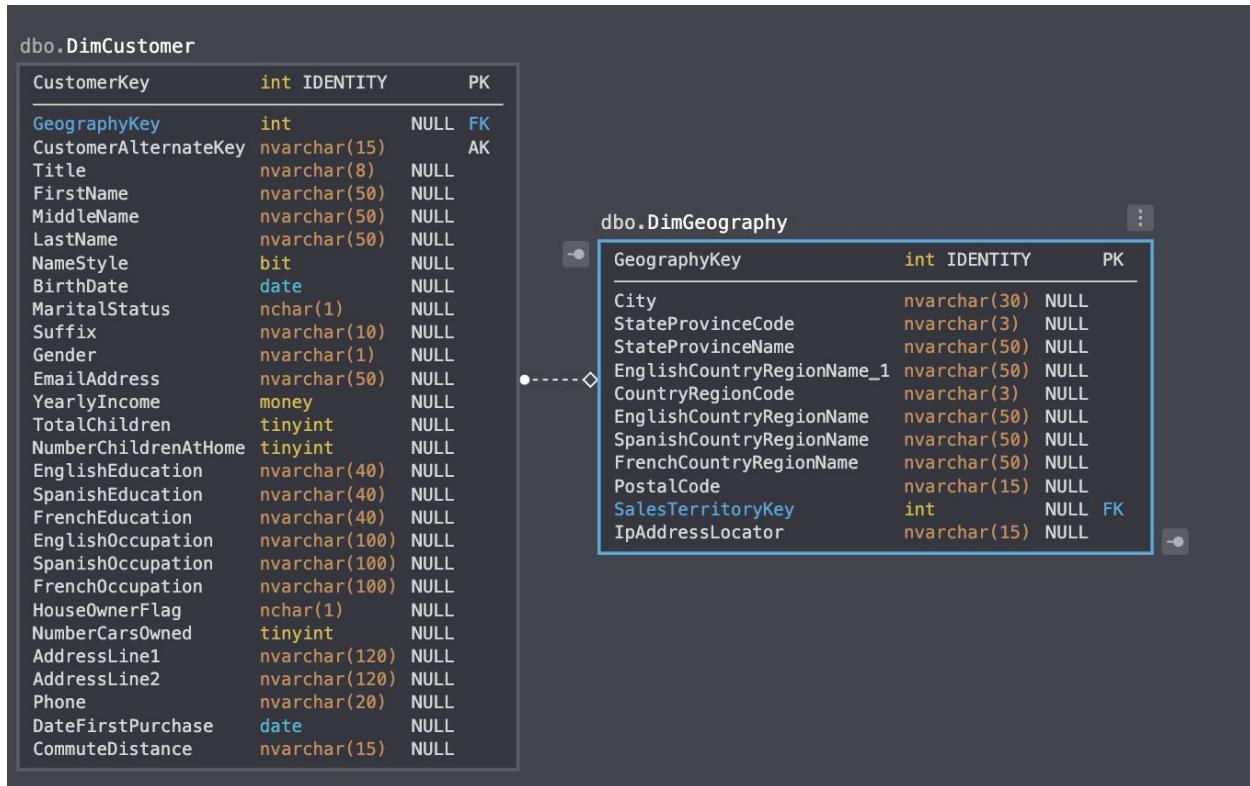
Diagrams of Tables

Key View



DATABASE PROPOSITIONS AND SOLUTIONS

Standard View



Attributes

Table Name	Column Name
DimCustomer	AVG_NUM_OF_CARS
DimGeography	City

Problem-solving Query

```
USE AdventureWorksDW2014
```

```
GO
```

```

SELECT TOP 67 b.City
      ,avg(a.NumberCarsOwned) AS [AVG_NUM_OF_CARS]
FROM DimCustomer AS a
JOIN DimGeography AS b ON a.GeographyKey = b.GeographyKey
GROUP BY B.City
    
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	City	AVG_NUM_OF_CARS
1	Ballard	1
2	Barstow	1
3	Basingstoke Hants	0
4	Baytown	2
5	Beaverton	1
6	Bell Gardens	1
7	Bellevue	2
8	Bellflower	1
9	Bellingham	1
10	Bendigo	1

Ln 97, Col 1 (201 selected) Spaces: 4 UTF-8 LF SQL MSSQL 67 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('City')
```

DATABASE PROPOSITIONS AND SOLUTIONS

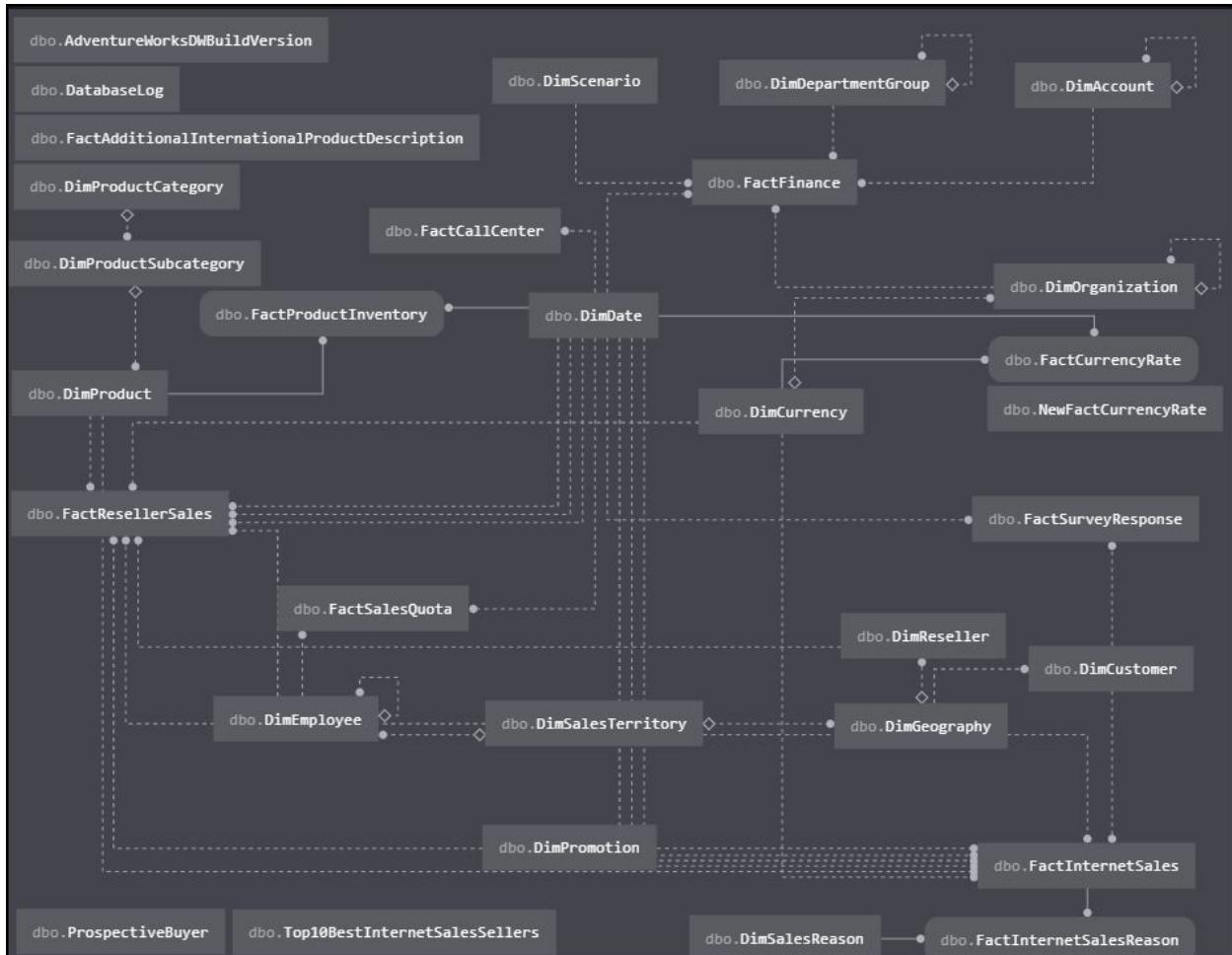
```
1  [
2      "City": [
3          {
4              "City": "Ballard",
5              "AVG_NUM_OF_CARS": 1
6          },
7          {
8              "City": "Barstow",
9              "AVG_NUM_OF_CARS": 1
10         },
11         {
12             "City": "Basingstoke Hants",
13             "AVG_NUM_OF_CARS": 0
14         },
15         {
16             "City": "Baytown",
17             "AVG_NUM_OF_CARS": 2
18         },
19         {
20             "City": "Beaverton",
21             "AVG_NUM_OF_CARS": 1
22         },
23         {
24             "City": "Bell Gardens",
25             "AVG_NUM_OF_CARS": 1
26         },
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 5

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Problem Statement

Group each category number by their english names

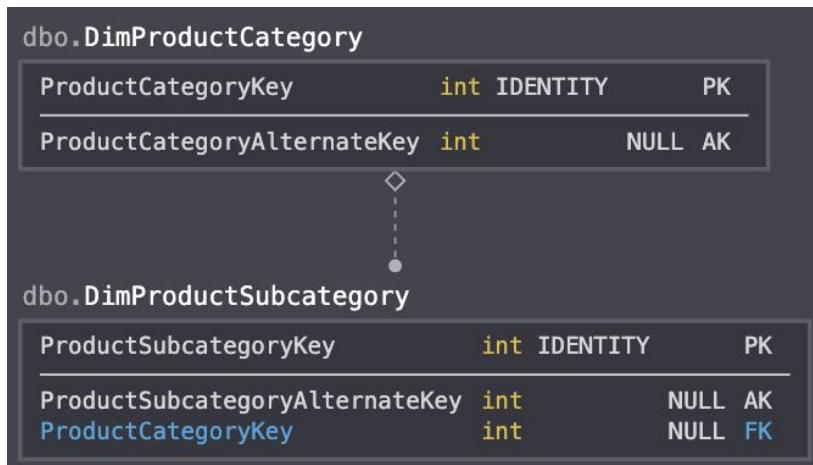
Database

AdventureWorksDW2014

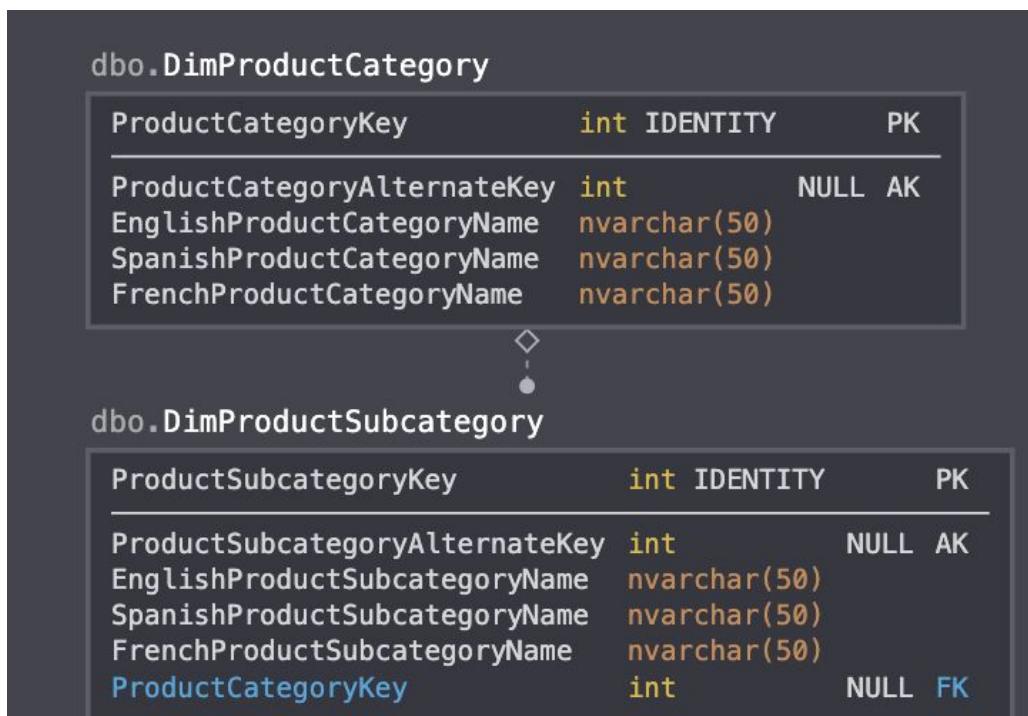
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
DimProductCategory	Product Category Num
DimProductSubcategory	EnglishProductCategoryName

Problem-solving Query

```
USE AdventureWorksDW2014
GO
SELECT TOP 23 avg(a.ProductCategoryKey) AS [Product Category Num]
    ,b.EnglishProductCategoryName
FROM DimProductSubCategory AS a
JOIN DimProductCategory AS b ON a.ProductCategoryKey = b.ProductCategoryKey
GROUP BY EnglishProductCategoryName
```

Sample Relational Output

	Product Category Num	EnglishProductCategoryName
1	4	Accessories
2	1	Bikes
3	3	Clothing
4	2	Components

Ln 115, Col 33 Spaces: 4 UTF-8 LF SQL MSSQL 4 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Product')
```

DATABASE PROPOSITIONS AND SOLUTIONS

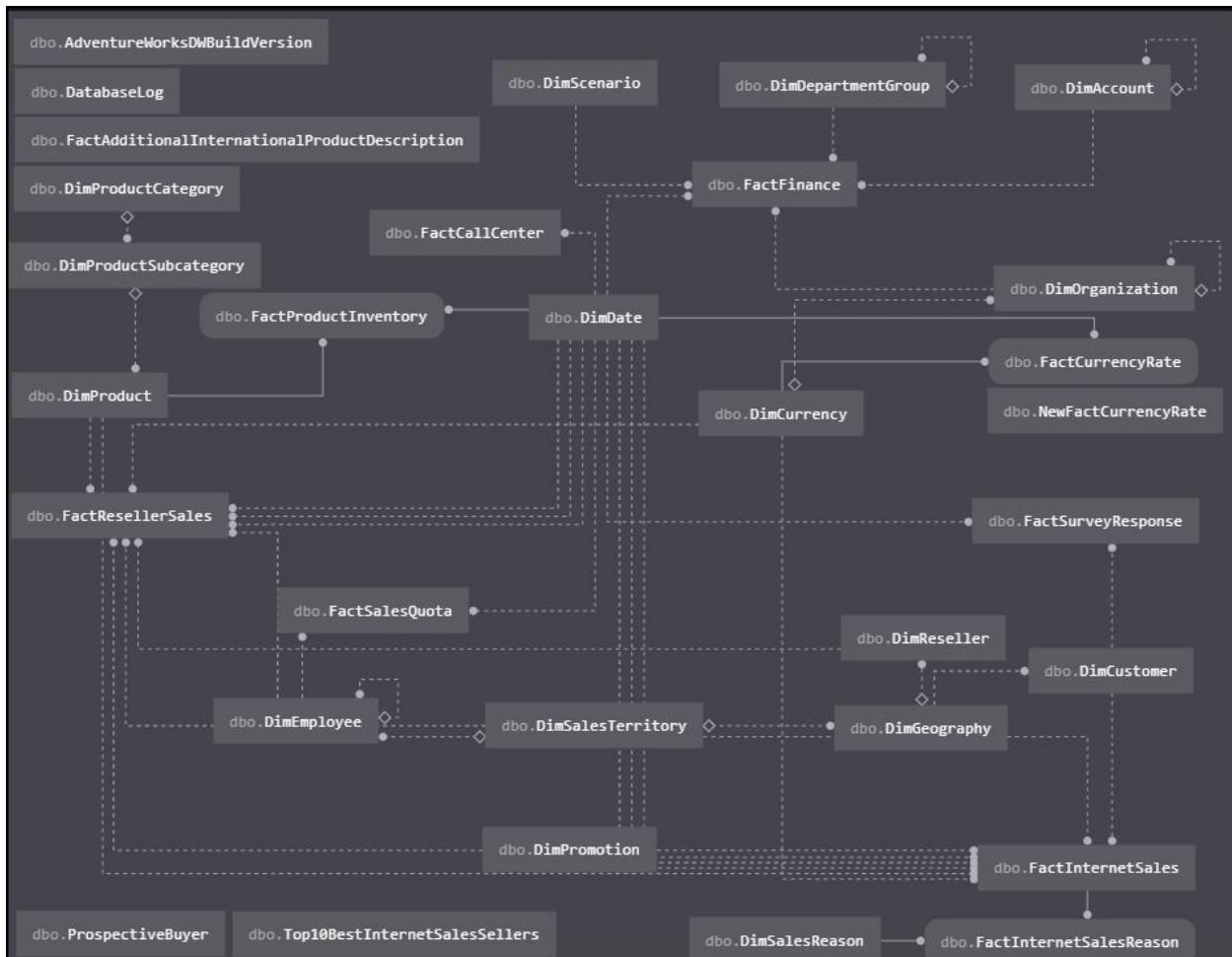
```
1  [
2      "Product": [
3          {
4              "Product Category Num": 4,
5              "EnglishProductName": "Accessories"
6          },
7          {
8              "Product Category Num": 1,
9              "EnglishProductName": "Bikes"
10         },
11         {
12             "Product Category Num": 3,
13             "EnglishProductName": "Clothing"
14         },
15         {
16             "Product Category Num": 2,
17             "EnglishProductName": "Components"
18         }
19     ]
20 ]
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 6

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Problem Statement

Find the average yearly revenue per 40 cities

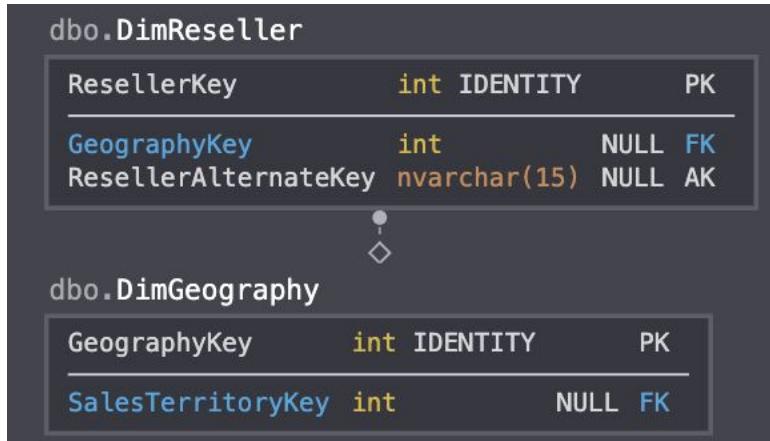
Database

AdventureWorksDW2014

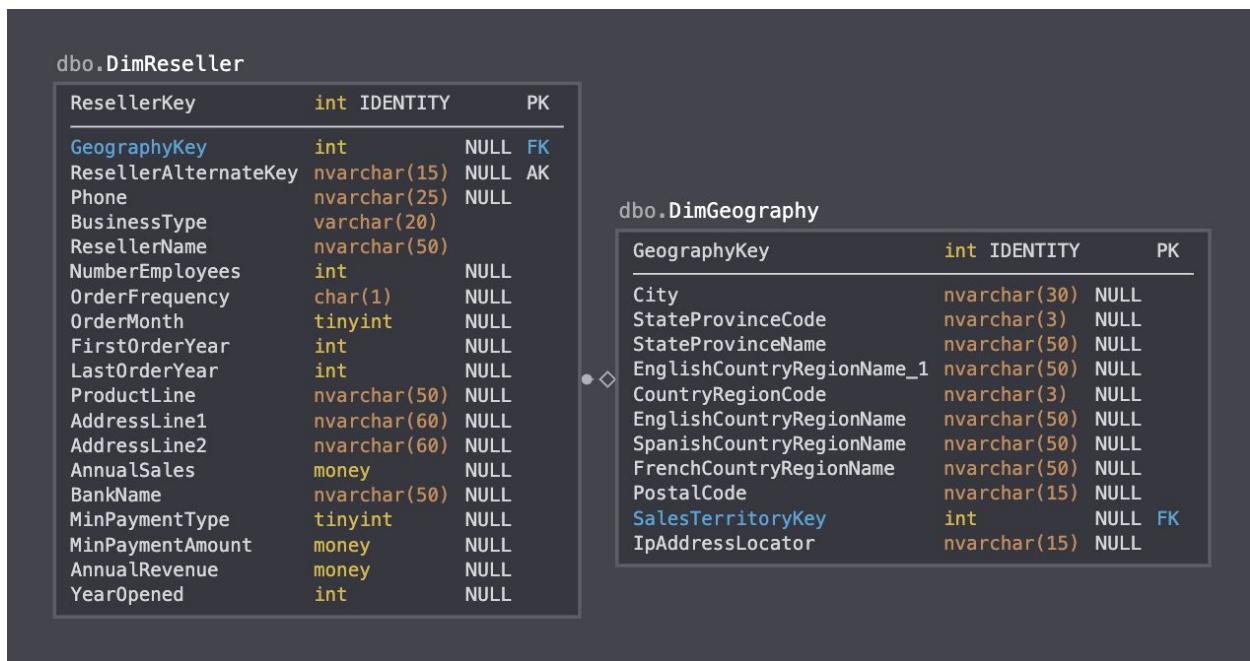
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



Attributes

Table Name	Column Name
DimReseller	AnnualRevenue
DimGeography	City

DATABASE PROPOSITIONS AND SOLUTIONS

Problem-solving Query

```
USE AdventureWorksDW2014
GO

SELECT TOP 40 AVG(AnnualRevenue) AS [AnnualRevenue]
      ,b.City
FROM DimReseller AS a
JOIN DimGeography AS b ON a.GeographyKey = b.GeographyKey
GROUP BY b.City
```

Sample Relational Output

	AnnualRevenue	City
1	150000.0000	Abingdon
2	115000.0000	Albany
3	165000.0000	Alexandria
4	150000.0000	Alhambra
5	100000.0000	Alpine
6	80000.0000	Altamonte Springs
7	300000.0000	Arlington
8	150000.0000	Ascheim
9	30000.0000	Atlanta
10	30000.0000	Auburn

Ln 115, Col 4 (29 selected) Spaces: 4 UTF-8 LF SQL MSSQL 40 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('City')
```

DATABASE PROPOSITIONS AND SOLUTIONS

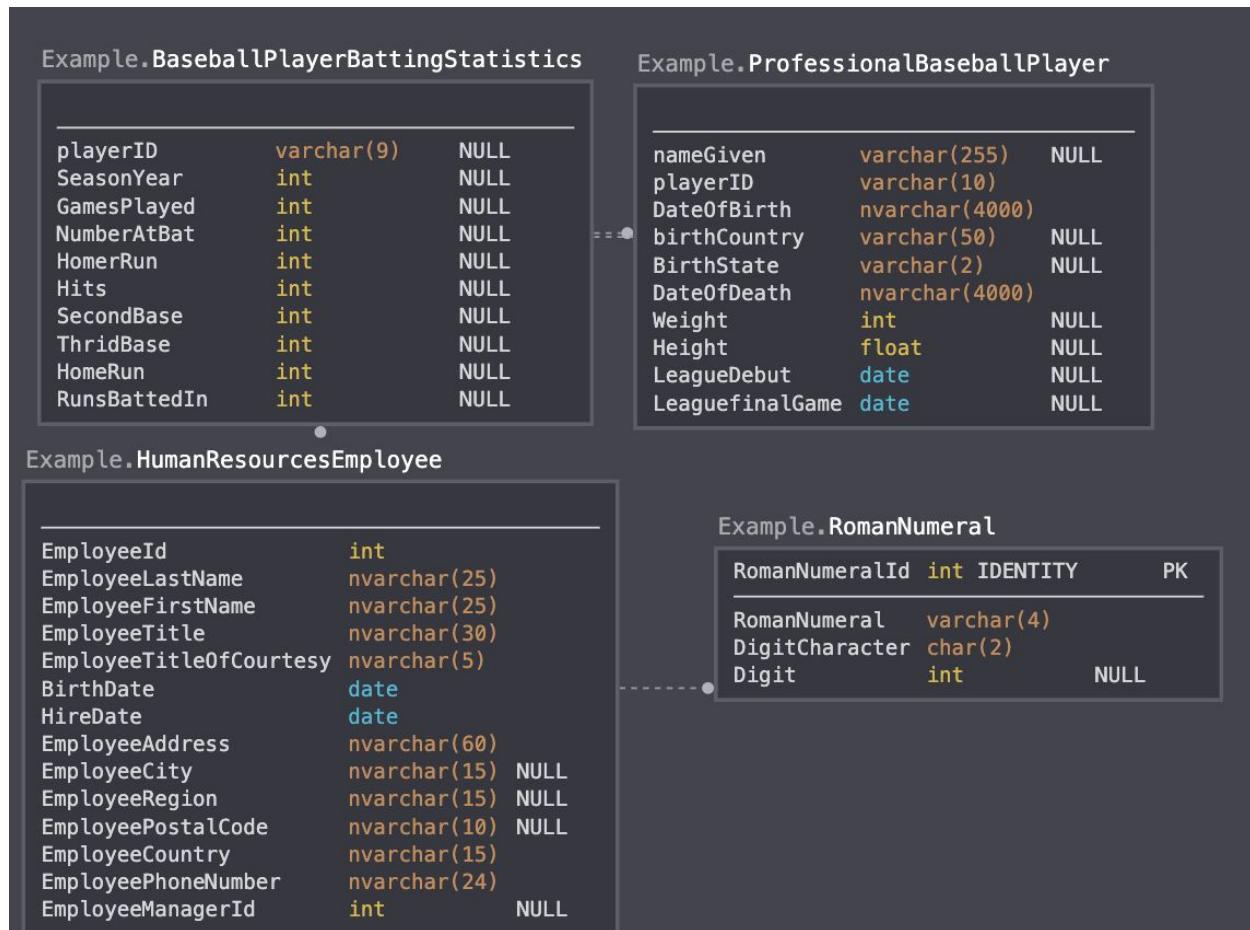
```
1  [
2      "City": [
3          {
4              "AnnualRevenue": 150000.0000,
5              "City": "Abingdon"
6          },
7          {
8              "AnnualRevenue": 115000.0000,
9              "City": "Albany"
10         },
11         {
12             "AnnualRevenue": 165000.0000,
13             "City": "Alexandria"
14         },
15         {
16             "AnnualRevenue": 150000.0000,
17             "City": "Alhambra"
18         },
19         {
20             "AnnualRevenue": 100000.0000,
21             "City": "Alpine"
22         },
23         {
24             "AnnualRevenue": 80000.0000,
25             "City": "Altamonte Springs"
26         }
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 7

Diagram of Database

Example subsystem in Northwinds



Problem Statement

Write a query that calculates the average percent markup of finished products by product subcategory.

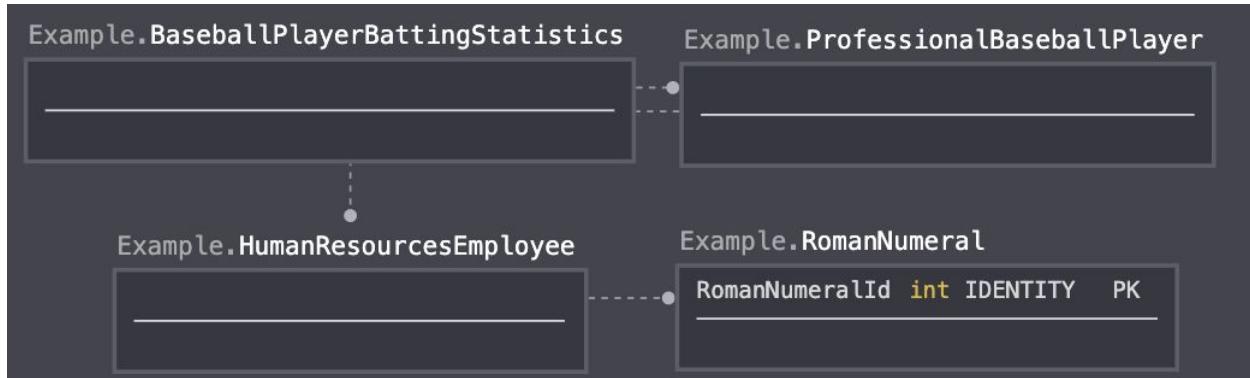
Database

Northwinds2019TSQLV5

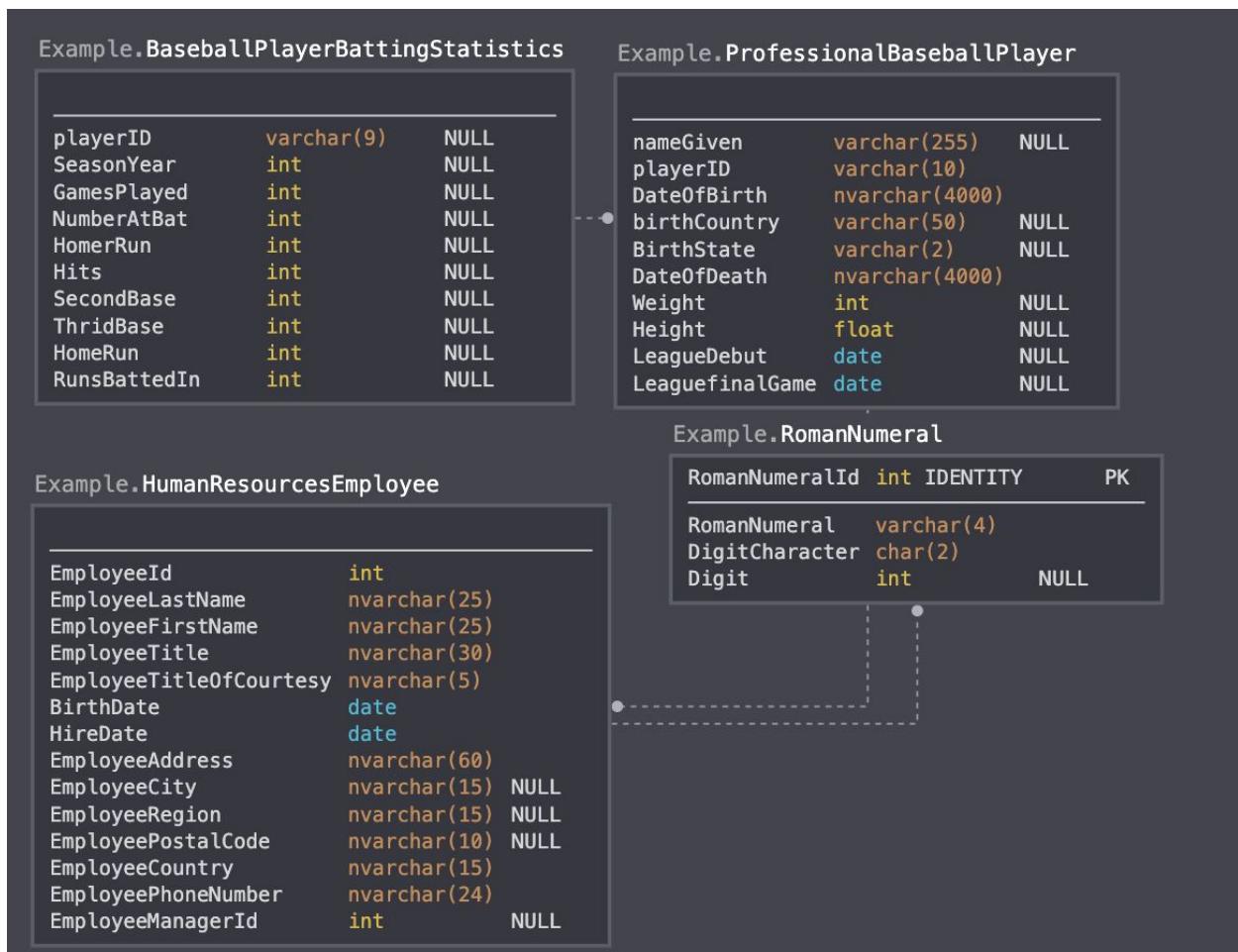
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
Example.BaseballPlayerBattingStatistics	NUM_OF_HOMERUNS
Example.ProfessionalBaseballPlayer	Weight

Order by

Table Name	Column Name	Sort Order
Example.ProfessionalBaseballPlayer	Weight	ASC

Problem-solving Query

```
USE Northwinds2019TSQLV5
GO
SELECT avg(a.homerun) AS [NUM_OF_HOMERUNS]
      ,b.Weight
FROM Example.BaseballPlayerBattingStatistics AS a
JOIN Example.ProfessionalBaseballPlayer AS b ON a.playerID = b.playerID
WHERE b.weight BETWEEN 150 AND 300
GROUP BY Weight
ORDER BY Weight
```

Sample Relational Output

	NUM_OF_HOMERUNS	Weight
1	1	150
2	0	151
3	0	152
4	2	153
5	0	154
6	1	155
7	0	156
8	0	157
9	0	158
10	0	159

Ln 125, Col 4 (8 selected) Spaces: 4 UTF-8 LF SQL MSSQL 104 rows

DATABASE PROPOSITIONS AND SOLUTIONS

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Homeruns')
1   [
2     "Homeruns": [
3       {
4         "NUM_OF_HOMERUNS": 1,
5         "Weight": 150
6       },
7       {
8         "NUM_OF_HOMERUNS": 0,
9         "Weight": 151
10      },
11      {
12        "NUM_OF_HOMERUNS": 0,
13        "Weight": 152
14      },
15      {
16        "NUM_OF_HOMERUNS": 2,
17        "Weight": 153
18      },
19      {
20        "NUM_OF_HOMERUNS": 0,
21        "Weight": 154
22      },
23      {
24        "NUM_OF_HOMERUNS": 1,
25        "Weight": 155
26      },

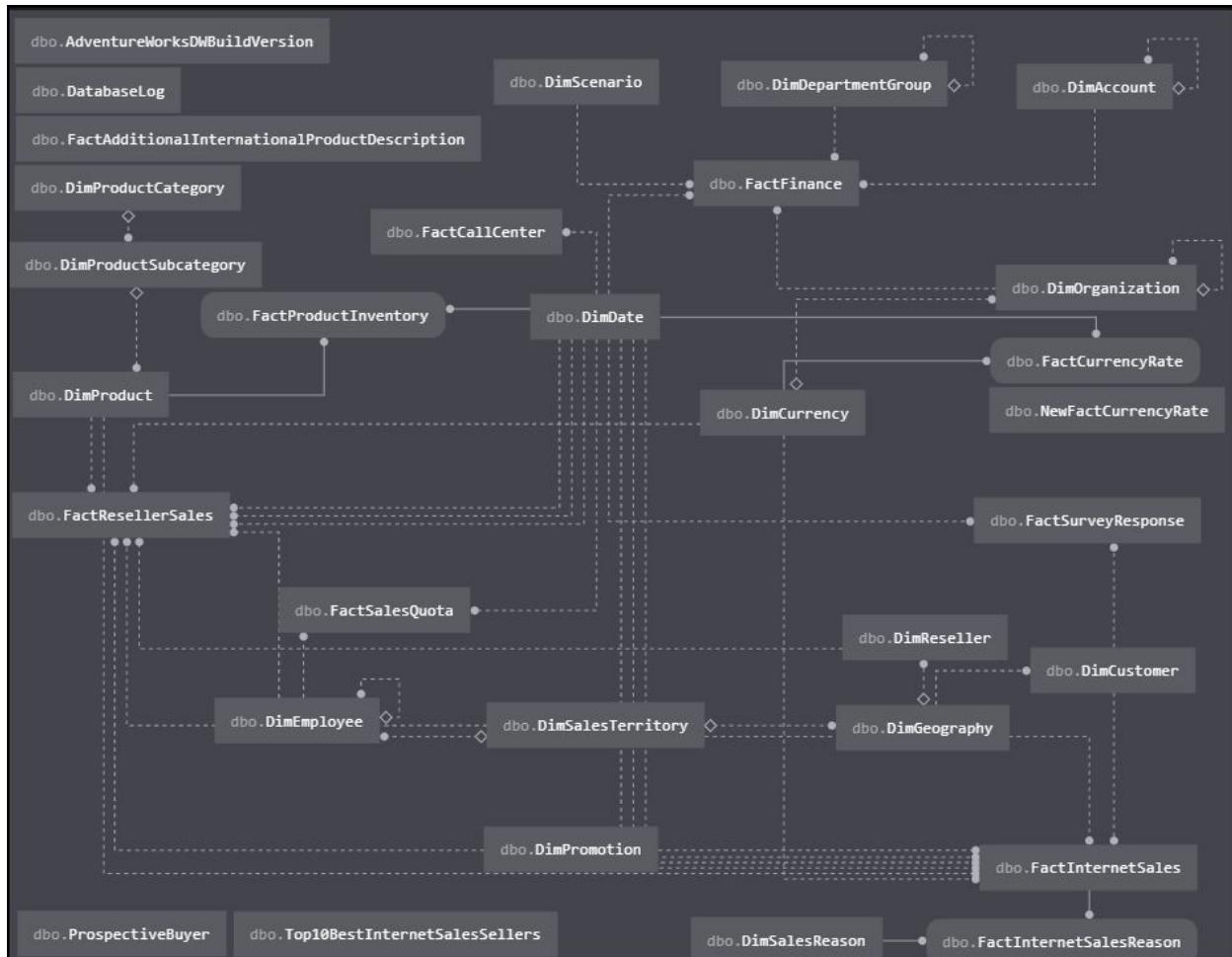
```

Proposition 8

Diagram of Database

dbo subsystem in AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS



Problem Statement

FIND THE AVERAGE TAX COST PER PRODUCT, ITS UNIT COST, AND CULTURE

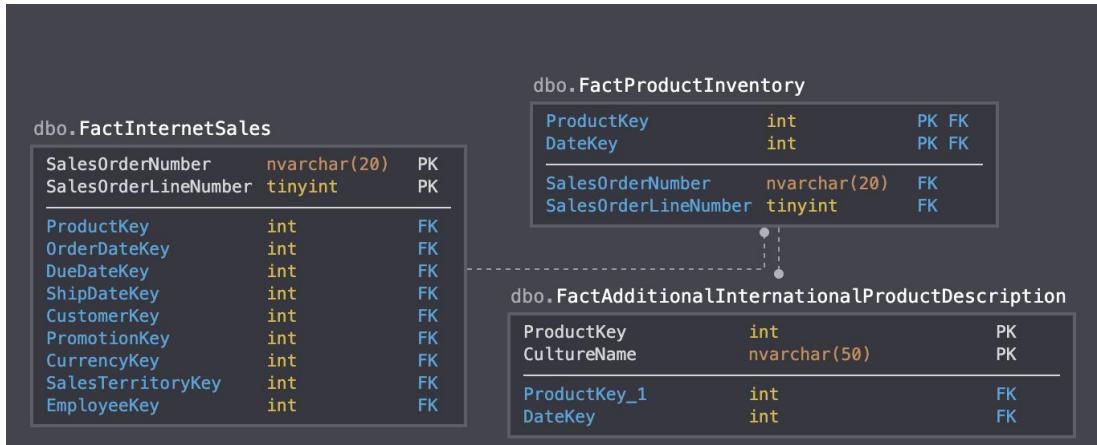
Database

AdventureWorksDW2014

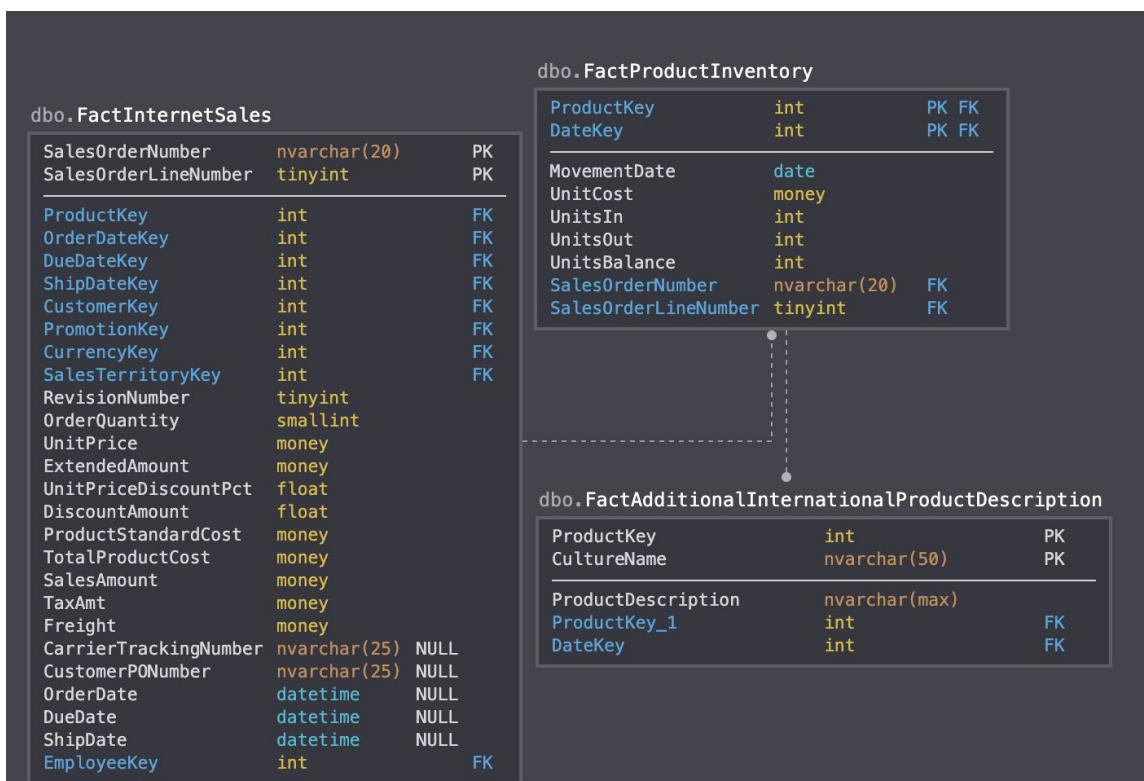
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
FactInternetSales	AVG_TAX_AMOUNT
FactProductInventory	UnitCost
FactAdditionalInternationalProductDescription	culture type

Problem-solving Query

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[ucsss] (
    @p NVARCHAR(87)
)
returns NVARCHAR(199)
BEGIN
    select @p = case
        when @p = 'Afrikaans' then 'African'
        end;
    RETURN @p
END
GO

USE AdventureWorksDW2014
GO

SELECT TOP 100 AVG(a.taxamt) AS [AVG_TAX_AMOUNT]
    ,b.UnitCost
    ,dbo.ucsss(c.CultureName) AS [culture type]
FROM FactInternetSales AS a
JOIN FactProductInventory AS b ON a.ProductKey = b.ProductKey
JOIN dbo.FactAdditionalInternationalProductDescription AS c ON b.ProductKey =
c.ProductKey
GROUP BY a.ProductKey
    ,b.UnitCost
    ,c.CultureName
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	AVG_TAX_AMOUNT	UnitCost	culture type
1	2.7992	8.9100	African
2	2.7992	8.9200	African
3	2.7992	8.9300	African
4	2.7992	8.9500	African
5	2.7992	8.9600	African
6	2.7992	8.9700	African
7	2.7992	8.9800	African
8	2.7992	8.9900	African
9	2.7992	9.0000	African
10	2.7992	9.0100	African

Ln 152, Col 3 Spaces: 4 UTF-8 LF SQL MSSQL 100 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Product')
```

DATABASE PROPOSITIONS AND SOLUTIONS

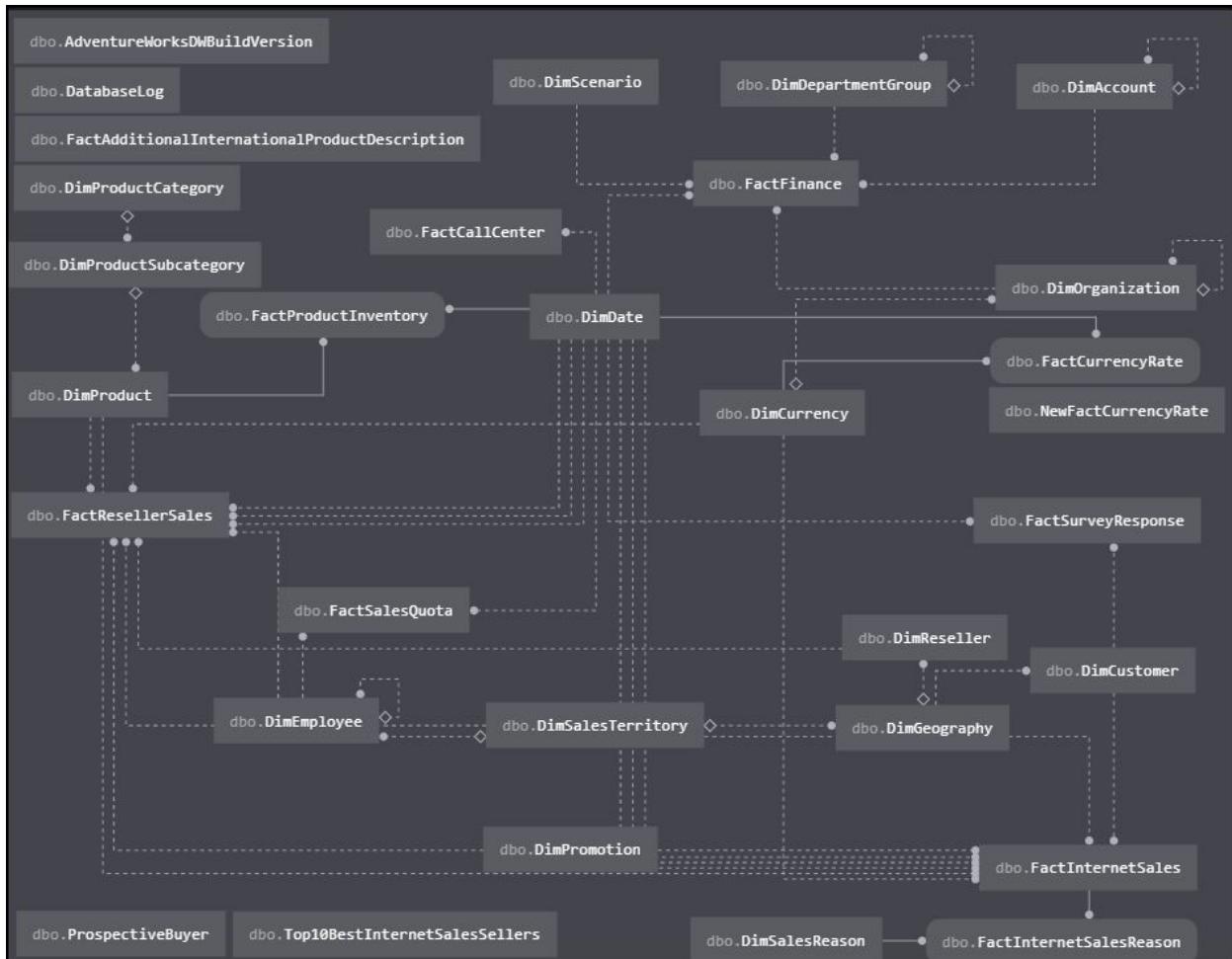
```
1  [
2      "Product": [
3          {
4              "AVG_TAX_AMOUNT": 2.7992,
5              "UnitCost": 8.9100,
6              "culture type": "African"
7          },
8          {
9              "AVG_TAX_AMOUNT": 2.7992,
10             "UnitCost": 8.9200,
11             "culture type": "African"
12         },
13         {
14             "AVG_TAX_AMOUNT": 2.7992,
15             "UnitCost": 8.9300,
16             "culture type": "African"
17         },
18         {
19             "AVG_TAX_AMOUNT": 2.7992,
20             "UnitCost": 8.9500,
21             "culture type": "African"
22         },
23         {
24             "AVG_TAX_AMOUNT": 2.7992,
25             "UnitCost": 8.9600,
26             "culture type": "African"
27         }
]
```

Proposition 9

Diagram of Database

dbo subsystem in AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS



Problem Statement

Find total sales for each department in every organization

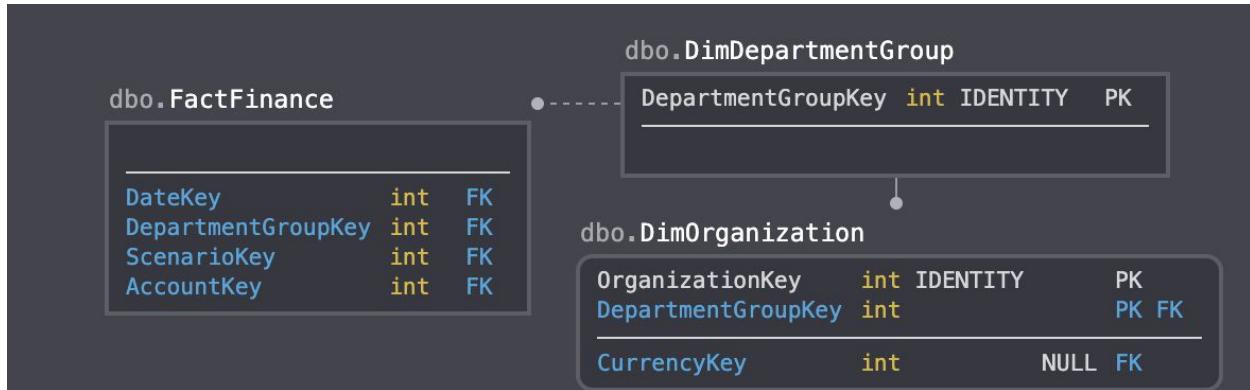
Database

AdventureWorksDW2014

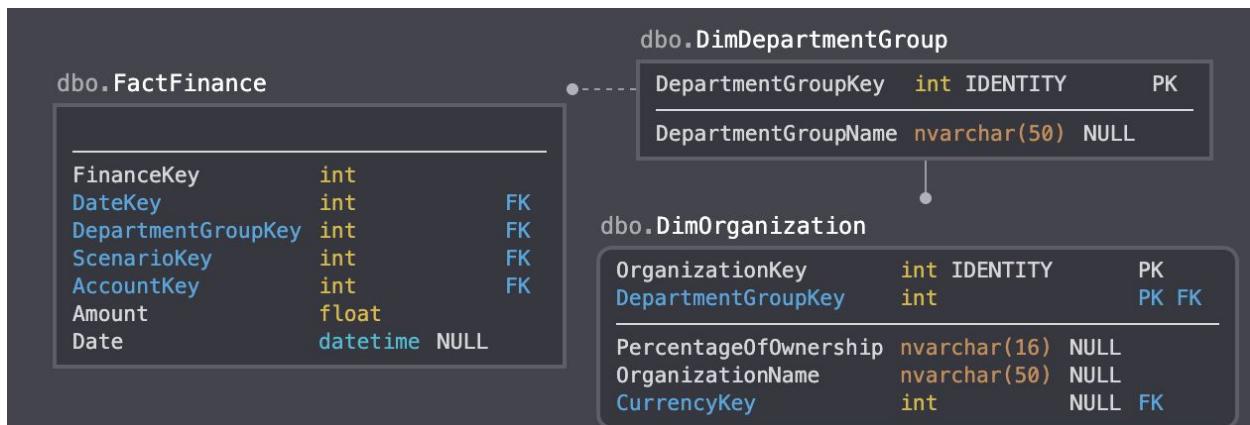
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key Views



Standard View



Attributes

Table Name	Column Name
FactFinance	TOTAL SALES OrganizationKey
DimDepartmentGroup	DepartmentGroupName myEvaluation
DimOrganization	OrganizationName

Problem-solving Query

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[AMTTT] (

```

Prepared by: Ibrahim Suhail

Date Prepared: 3/21/2020

C:\Users\Ibrahim Suhail\Desktop\Spring 2020\CS331\Project 1\G12-5 - Ibrahim Suhail.docx

Page 44 of 88

DATABASE PROPOSITIONS AND SOLUTIONS

```

    @p NVARCHAR(100)
)
returns NVARCHAR(1990)
BEGIN
    declare @res NVARCHAR(100)
    select @res = case
        when @p = 'corporate' then 'GREAT WORK'
        when @p like 'Exec%' then 'AMAZING JOB'
        ELSE 'You should be ashamed of yourself'
    end;
    RETURN @res
END
GO

USE AdventureWorksDW2014
GO

SELECT sum(a.Amount) AS [TOTAL SALES]
    ,a.OrganizationKey
    ,b.DepartmentGroupName
    ,c.OrganizationName
    ,dbo.amttt(DepartmentGroupName) AS [myEvaluation]
FROM FactFinance AS a
JOIN DimDepartmentGroup AS b ON a.DepartmentGroupKey = b.DepartmentGroupKey
JOIN DimOrganization AS c ON a.OrganizationKey = c.OrganizationKey
GROUP BY a.OrganizationKey
    ,b.DepartmentGroupName
    ,c.OrganizationName
ORDER BY DepartmentGroupName

```

Sample Relational Output

	TOTAL SALES	OrganizationKey	DepartmentGroupName	OrganizationName	myEvaluation
1	3569369	7	Corporate	Southwest Division	GREAT WORK
2	3714323	4	Corporate	Northwest Division	GREAT WORK
3	1494237	6	Corporate	Southeast Division	GREAT WORK
4	873980.69	11	Corporate	France	GREAT WORK
5	363443.43	13	Corporate	Australia	GREAT WORK
6	3453530	3	Corporate	Northeast Division	GREAT WORK
7	4399861.60999999	8	Corporate	Canadian Division	GREAT WORK
8	2782452	5	Corporate	Central Division	GREAT WORK
9	402727.88	12	Corporate	Germany	GREAT WORK
10	964953.06	12	Executive General and Adm...	Germany	AMAZING JOB

Ln 173, Col 73 Spaces: 4 UTF-8 LF SQL MSSQL 42 rows

DATABASE PROPOSITIONS AND SOLUTIONS

Sample JSON Output

Add below line 7:

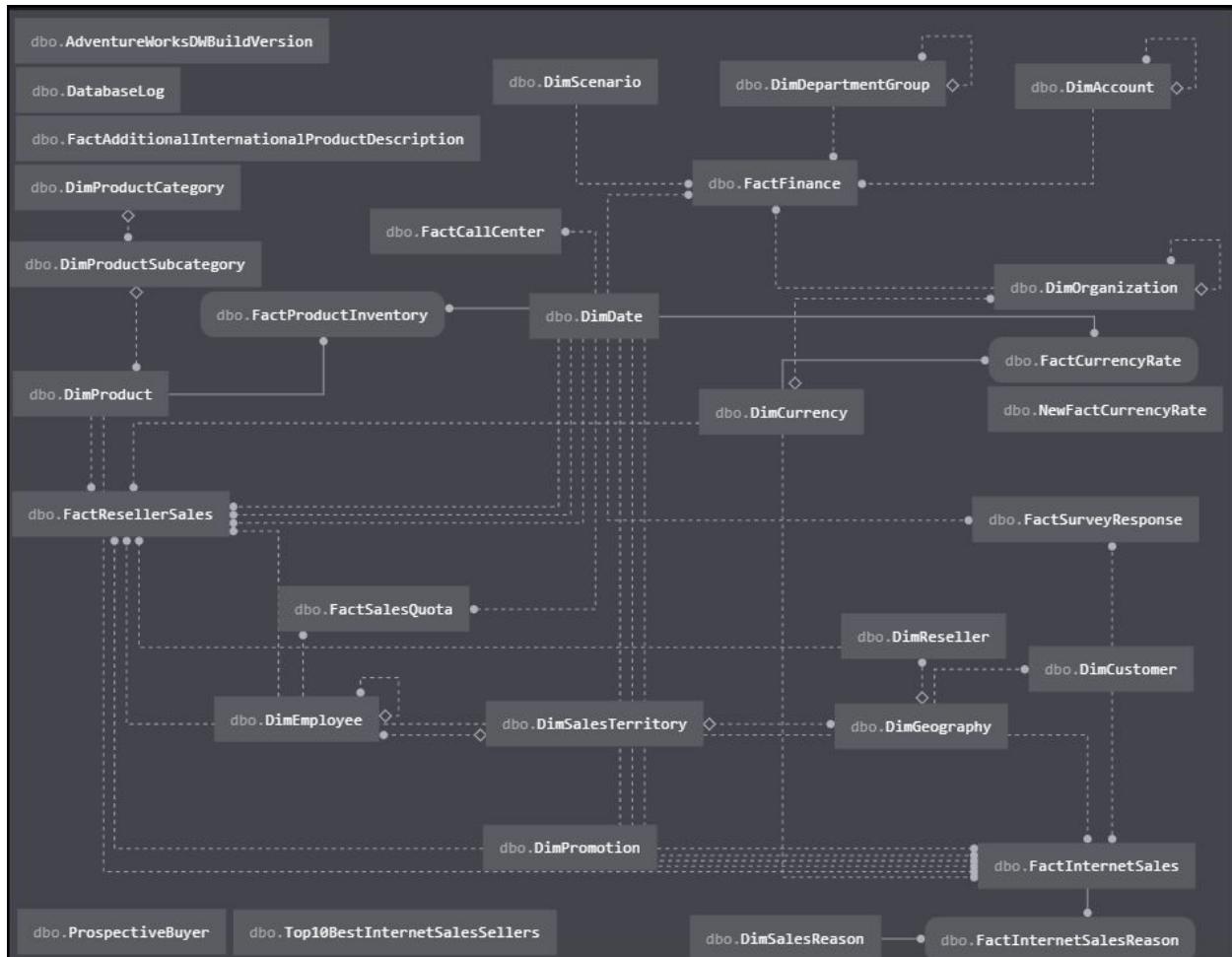
```
for json PATH, root('Department')  
  
1  [{  
2      "Department": [  
3          {  
4              "TOTAL SALES": 3.569369000000000e+006,  
5              "OrganizationKey": 7,  
6              "DepartmentGroupName": "Corporate",  
7              "OrganizationName": "Southwest Division",  
8              "myEvaluation": "GREAT WORK"  
9          },  
10         {  
11             "TOTAL SALES": 3.714323000000000e+006,  
12             "OrganizationKey": 4,  
13             "DepartmentGroupName": "Corporate",  
14             "OrganizationName": "Northwest Division",  
15             "myEvaluation": "GREAT WORK"  
16         },  
17         {  
18             "TOTAL SALES": 1.494237000000000e+006,  
19             "OrganizationKey": 6,  
20             "DepartmentGroupName": "Corporate",  
21             "OrganizationName": "Southeast Division",  
22             "myEvaluation": "GREAT WORK"  
23         },  
24     ]  
25 }]
```

Proposition 10

Diagram of Database

dbo subsystem in AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS



Problem Statement

Find the top 100 product keys, their sale dates, and which regions buy them the most

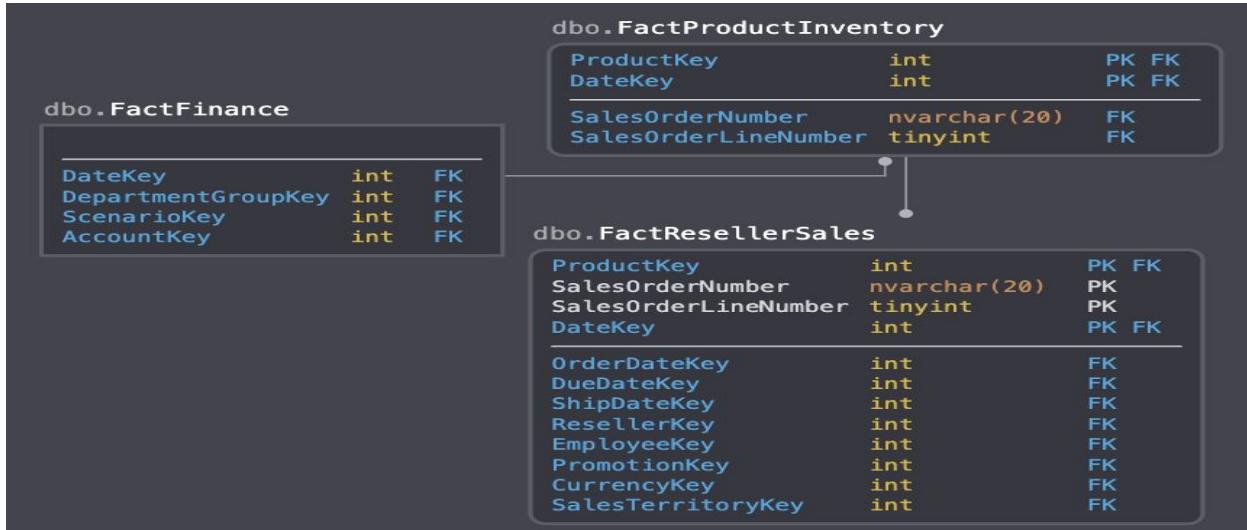
Database

AdventureWorksDW2014

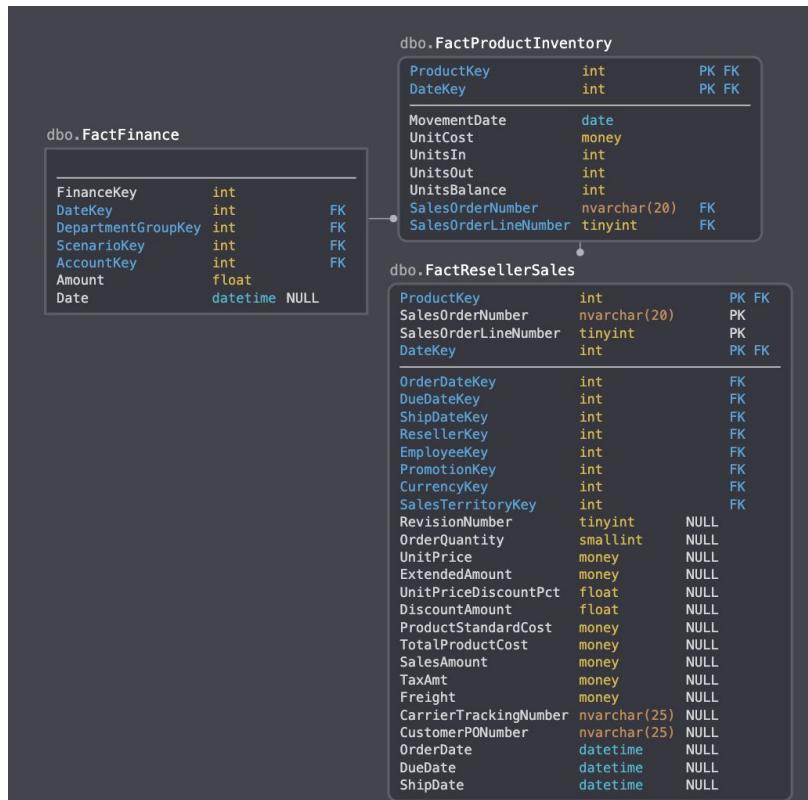
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
FactFinance	DateKey
FactProductInventory	ProductKey
FactResellerSales	AVG SALE REGION ROMAN_NUMERAL

Problem-solving Query

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create FUNCTION [dbo].[rn] (
    @p INT
)
returns nvarCHAR(10)
begin
    declare @res NVARCHAR(10)
    SELECT @res = case
        when @p = '1' then 'I'
        when @p = '2' then 'II'
        when @p = '3' then 'III'
        when @p = '4' then 'IV'
        when @p = '5' then 'V'
        end;
    return @res
END;
GO

```

```

USE AdventureWorksDW2014
GO

SELECT TOP 100 a.DateKey
    ,b.ProductKey
    ,avg(c.salesterritorykey) AS [AVG SALE REGION]
    ,DBO.RN(avg(c.salesterritorykey)) AS [ROMAN_NUMERAL]
FROM FactFinance AS a
JOIN FactProductInventory AS b ON a.DateKey = b.DateKey
JOIN FactResellerSales AS c ON b.ProductKey = c.ProductKey
GROUP BY a.DateKey
    ,b.ProductKey

```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	DateKey	ProductKey	Avg Sale Region	Roman Numeral
1	20101229	212	3	III
2	20110129	212	3	III
3	20110301	212	3	III
4	20110331	212	3	III
5	20110501	212	3	III
6	20110531	212	3	III
7	20110701	212	3	III
8	20110801	212	3	III
9	20110829	212	3	III
10	20110929	212	3	III

Ln 180, Col 29 Spaces: 4 UTF-8 LF SQL MSSQL 100 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Product')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  {
2      "Product": [
3          {
4              "DateKey": 20101229,
5              "ProductKey": 212,
6              "AVG SALE REGION": 3,
7              "ROMAN_NUMERAL": "III"
8          },
9          {
10             "DateKey": 20110129,
11             "ProductKey": 212,
12             "AVG SALE REGION": 3,
13             "ROMAN_NUMERAL": "III"
14         },
15         {
16             "DateKey": 20110301,
17             "ProductKey": 212,
18             "AVG SALE REGION": 3,
19             "ROMAN_NUMERAL": "III"
20         },
21         {
22             "DateKey": 20110331,
23             "ProductKey": 212,
24             "AVG SALE REGION": 3,
25             "ROMAN_NUMERAL": "III"
26         },

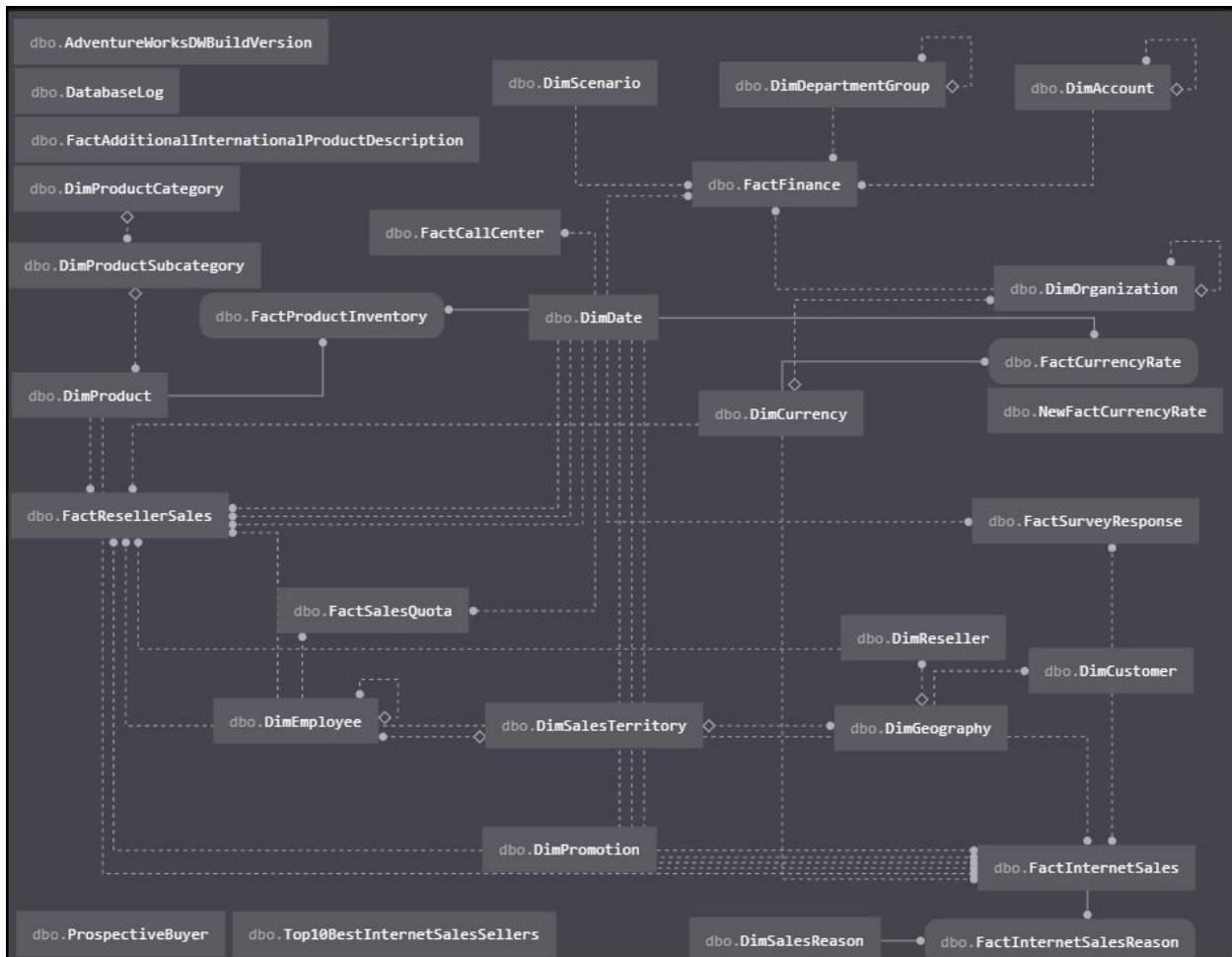
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 11

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Problem Statement

Find the average customer's income for each country, its name in French, and estimated hourly salary

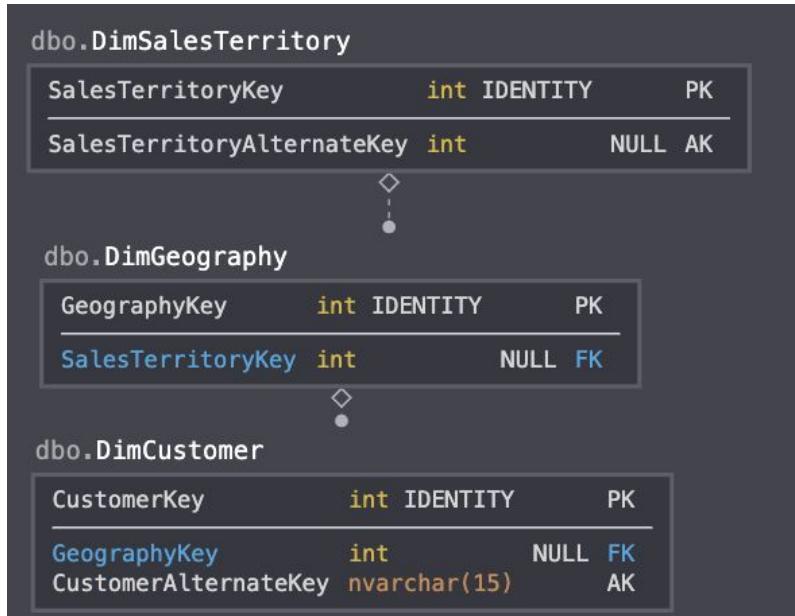
Database

AdventureWorksDW2014

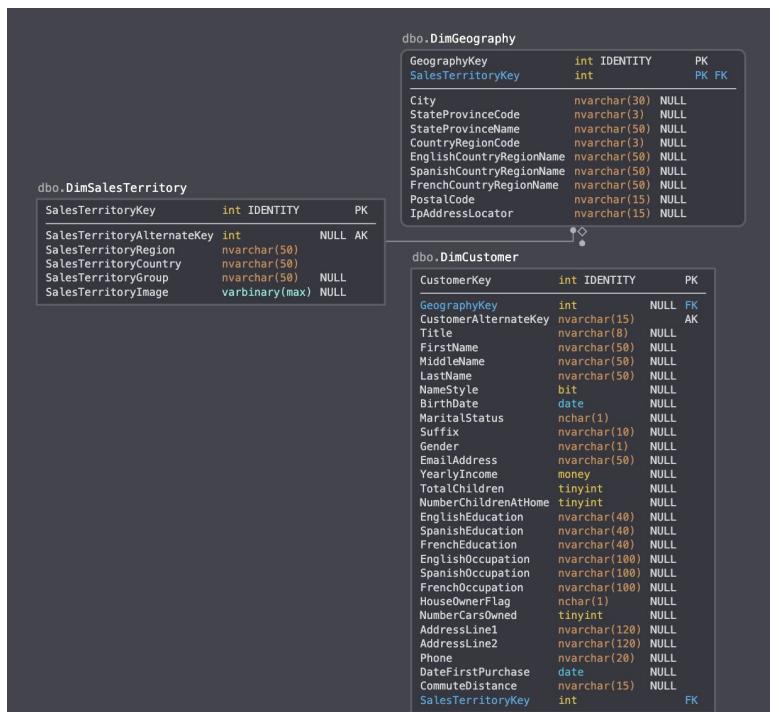
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
DimSalesTerritory	SalesTerritoryRegion
DimGeography	frenchCountryRegionName
DimCustomer	AVG_INCOME_PER_YEAR ESTIMATED HOURLY SALARY

Problem-solving Query

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create FUNCTION [dbo].[toHour] (
    @p INT
)
returns int
begin
    return @p / 8760
END;
GO

USE AdventureWorksDW2014
GO

SELECT a.SalesTerritoryRegion
      ,b.frenchCountryRegionName
      ,AVG(c.YearlyIncome) AS [AVG_INCOME_PER_YEAR]
      ,dbo.toHour(AVG(c.YearlyIncome)) AS [ESTIMATED HOURLY SALARY]
FROM DimSalesTerritory AS A
JOIN DimGeography AS b ON A.SalesTerritoryKey = b.SalesTerritoryKey
JOIN DimCustomer AS c ON b.GeographyKey = c.GeographyKey
GROUP BY a.SalesTerritoryRegion
      ,b.frenchCountryRegionName
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	SalesTerritoryRegion	frenchCountryRegionName	AVG_INCOME_PER_YEAR	ESTIMATED HOURLY SALARY
1	Central	États-Unis	53750.0000	6
2	Australia	Australie	64338.6243	7
3	Canada	Canada	57167.4092	6
4	Germany	Allemagne	42943.8202	4
5	Northeast	États-Unis	48750.0000	5
6	United Kingdom	Royaume-Uni	52169.3674	5
7	France	France	35762.4309	4
8	Northwest	États-Unis	59778.5094	6
9	Southeast	États-Unis	59166.6666	6
10	Southwest	États-Unis	66555.0561	7

Ln 186, Col 1 (420 selected) Spaces: 4 UTF-8 LF SQL MSSQL 10 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Customer')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  [
2      "Customer": [
3          {
4              "SalesTerritoryRegion": "Central",
5              "frenchCountryRegionName": "États-Unis",
6              "AVG_INCOME_PER_YEAR": 53750.0000,
7              "ESTIMATED HOURLY SALARY": 6
8          },
9          {
10             "SalesTerritoryRegion": "Australia",
11             "frenchCountryRegionName": "Australie",
12             "AVG_INCOME_PER_YEAR": 64338.6243,
13             "ESTIMATED HOURLY SALARY": 7
14         },
15         {
16             "SalesTerritoryRegion": "Canada",
17             "frenchCountryRegionName": "Canada",
18             "AVG_INCOME_PER_YEAR": 57167.4092,
19             "ESTIMATED HOURLY SALARY": 6
20         },
21         {
22             "SalesTerritoryRegion": "Germany",
23             "frenchCountryRegionName": "Allemagne",
24             "AVG_INCOME_PER_YEAR": 42943.8202,
25             "ESTIMATED HOURLY SALARY": 4
26         },

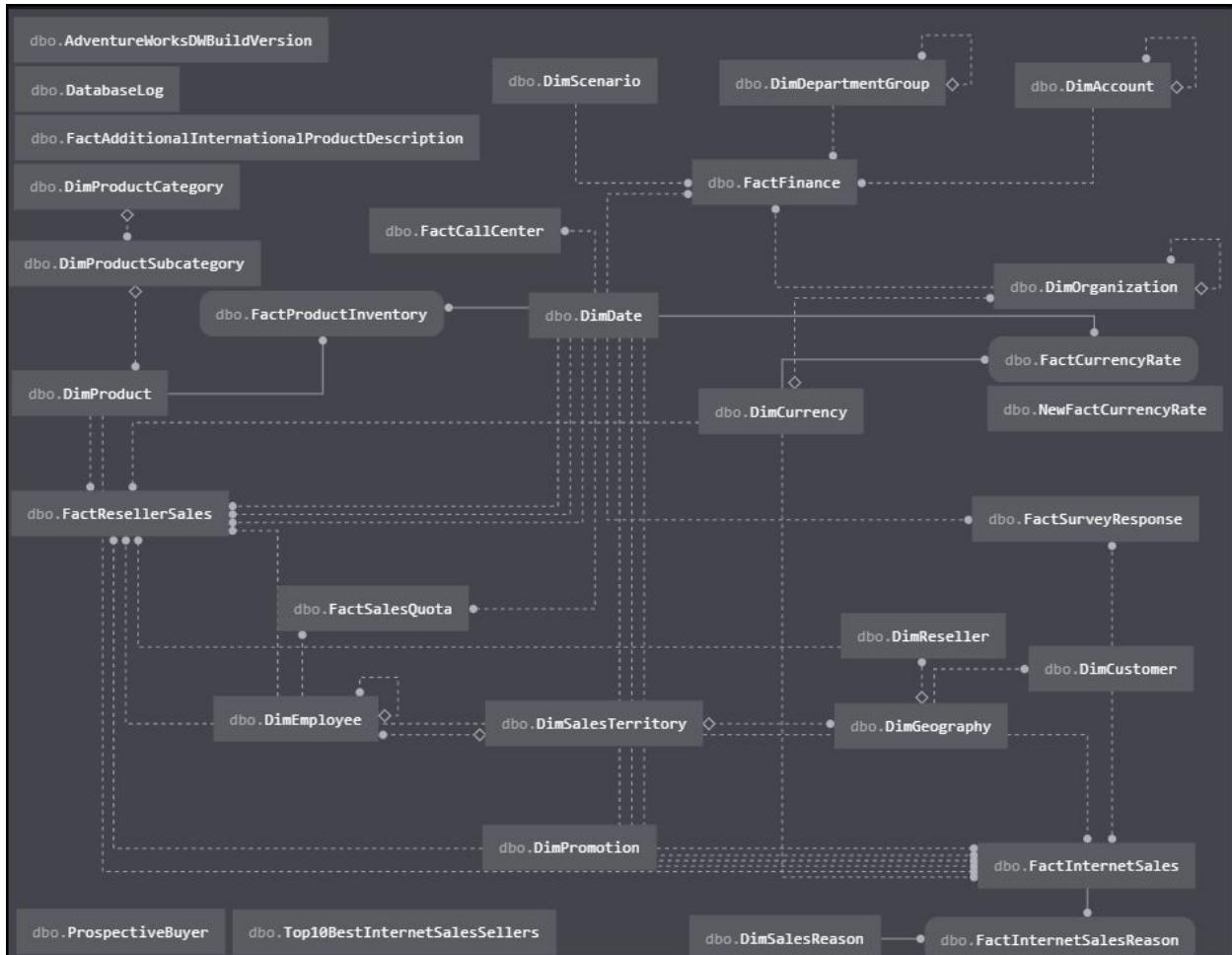
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 12

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Problem Statement

Find the lowest sales quotas, the customer who answered the survey about this employee, their email address and domain

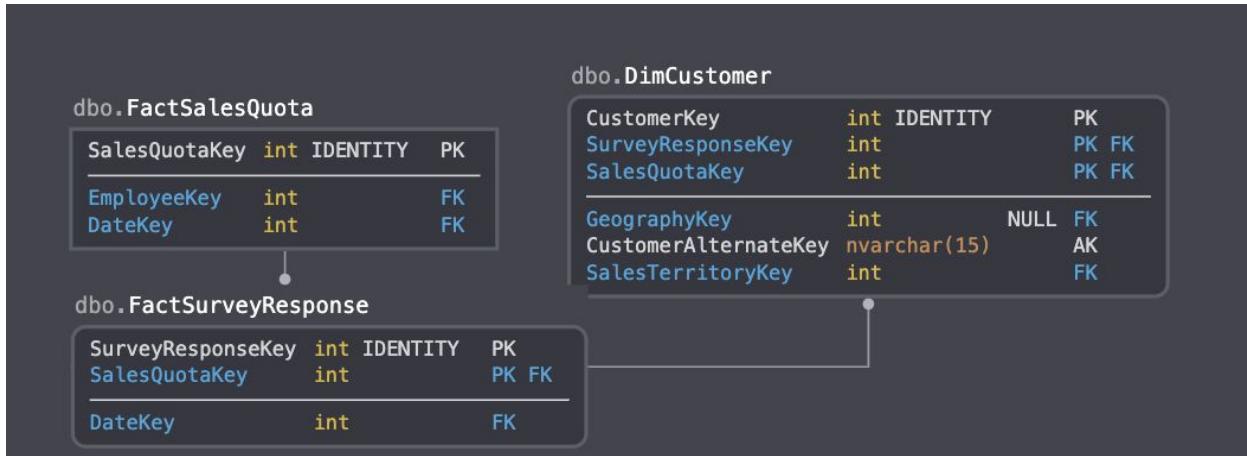
Database

AdventureWorksDW2014

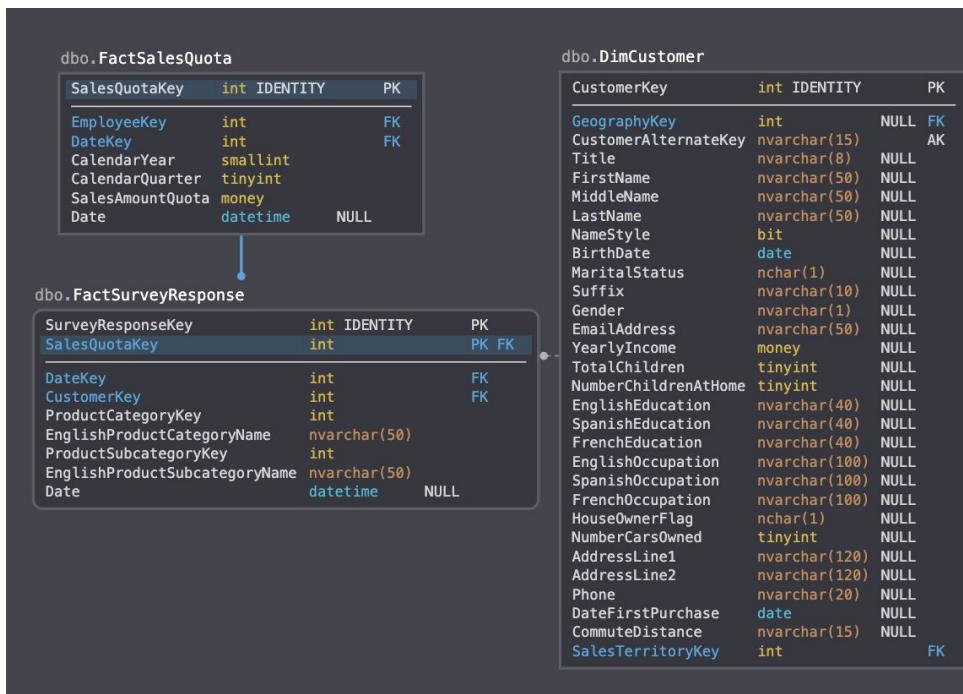
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

Table Name	Column Name
FactSalesQuota	min_sales_amt_qta
FactSurveyResponse	CustomerKey
DimCustomer	EmailAddress DOMAIN

Problem-solving Query

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[dom] (
    @p NVARCHAR(100)
)
returns NVARCHAR(180)
begin
    return SUBSTRING(@p, CHARINDEX('@', @p) + 1, LEN(@p))

END ;
GO

USE AdventureWorksDW2014
GO

SELECT TOP 50 min(a.SalesAmountQuota) AS [min_sales_amt_qta]
    ,b.
    ,c.EmailAddress
    ,dbo.dom(c.EmailAddress) AS [DOMAIN]
FROM FactSalesQuota AS a
JOIN FactSurveyResponse AS b ON a.DateKey = b.DateKey
JOIN DimCustomer AS c ON b.CustomerKey = c.CustomerKey
GROUP BY b.CustomerKey
    ,c.EmailAddress
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	min_sales_amt_qta	CustomerKey	EmailAddress	DOMAIN
1	15000.0000	11735	sydney16@adventure-works...	adventure-works.com
2	15000.0000	12056	ian69@adventure-works.com	adventure-works.com
3	15000.0000	13006	whitney10@adventure-works...	adventure-works.com
4	15000.0000	14264	karl9@adventure-works.com	adventure-works.com
5	15000.0000	14453	jesse37@adventure-works.c...	adventure-works.com
6	15000.0000	14988	mandy13@adventure-works.c...	adventure-works.com
7	15000.0000	15019	mary38@adventure-works.com	adventure-works.com
8	15000.0000	15124	wendy4@adventure-works.com	adventure-works.com
9	15000.0000	15145	crystal12@adventure-works...	adventure-works.com
10	15000.0000	15303	haley29@adventure-works.c...	adventure-works.com

Ln 205, Col 37 Spaces: 4 UTF-8 LF SQL MSSQL 30 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Customer')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  [
2      "Customer": [
3          {
4              "SalesTerritoryRegion": "Central",
5              "frenchCountryRegionName": "États-Unis",
6              "AVG_INCOME_PER_YEAR": 53750.0000,
7              "ESTIMATED HOURLY SALARY": 6
8          },
9          {
10             "SalesTerritoryRegion": "Australia",
11             "frenchCountryRegionName": "Australie",
12             "AVG_INCOME_PER_YEAR": 64338.6243,
13             "ESTIMATED HOURLY SALARY": 7
14         },
15         {
16             "SalesTerritoryRegion": "Canada",
17             "frenchCountryRegionName": "Canada",
18             "AVG_INCOME_PER_YEAR": 57167.4092,
19             "ESTIMATED HOURLY SALARY": 6
20         },
21         {
22             "SalesTerritoryRegion": "Germany",
23             "frenchCountryRegionName": "Allemagne",
24             "AVG_INCOME_PER_YEAR": 42943.8202,
25             "ESTIMATED HOURLY SALARY": 4
26         },

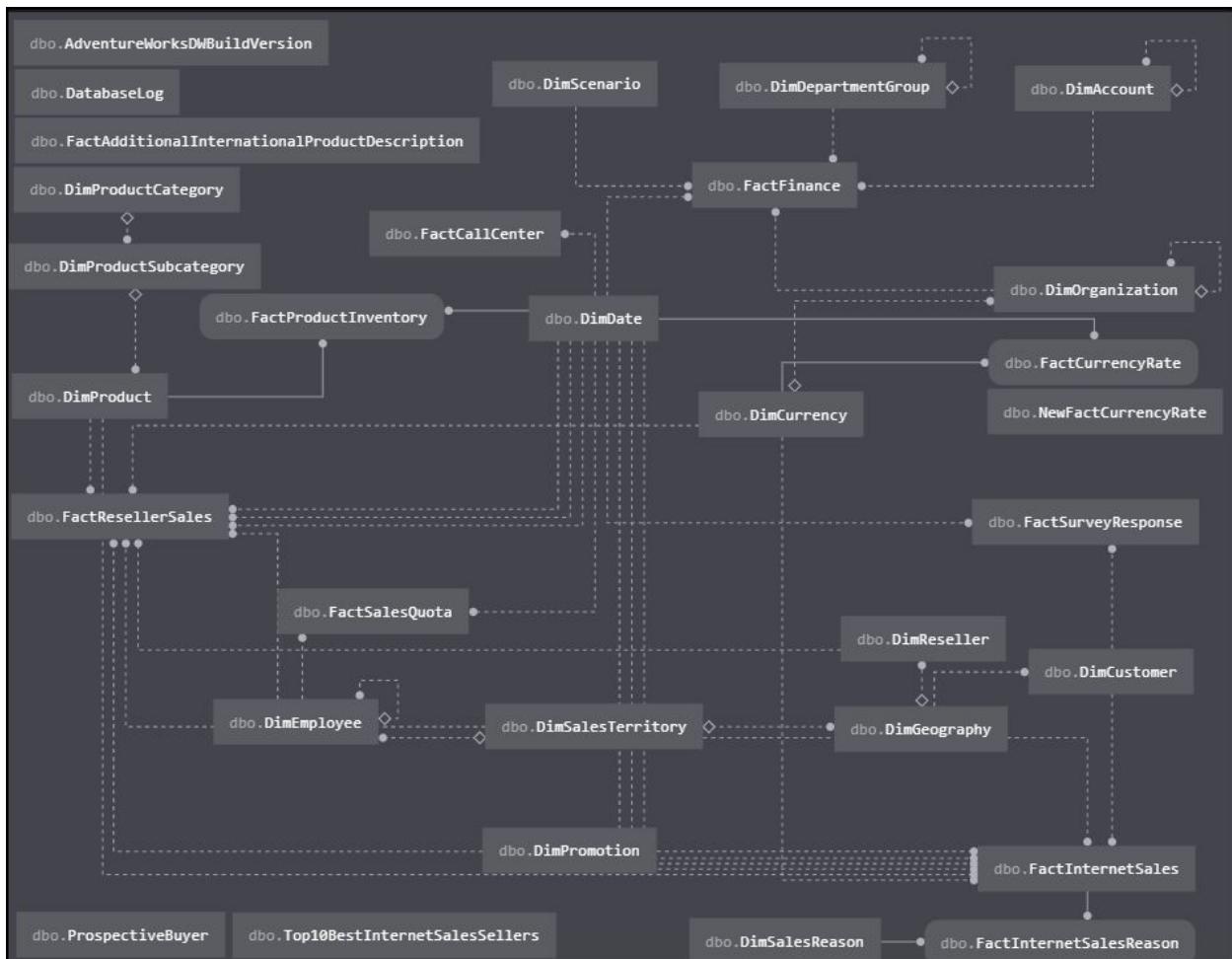
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 13

Diagram of Database

dbo subsystem in AdventureWorksDW2014



Problem Statement

Find the oldest customer, the model they ordered, and their ESTIMATED date of birth

Database

AdventureWorksDW2014

DATABASE PROPOSITIONS AND SOLUTIONS

Attributes

View Name	Column Name
vAssocSeqLineItems	Model CustomerContactName
vAssocSeqOrders	IncomeGroup OrderDate
vDMPrep	ESTIMATED BIRTHDATE OLDEST PERSON

Problem-solving Query

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create FUNCTION [dbo].[edob]
    @p int
)
returns int
BEGIN
RETURN year(dateadd(YEAR, -@p, getdate()))
END ;
GO

USE AdventureWorksDW2014
GO
SELECT TOP 46 MAX(c.Age) AS [OLDEST PERSON]
    ,a.Model
    ,b.IncomeGroup
    ,dbo.edob(MAX(c.Age)) AS [ESTIMATED BIRTHDATE]
FROM dbo.vAssocSeqLineItems AS a
JOIN dbo.vAssocSeqOrders AS b ON a.OrderNumber = b.OrderNumber
JOIN dbo.vDMPrep AS c ON b.OrderNumber = c.OrderNumber
GROUP BY a.Model
    ,b.IncomeGroup
    
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	OLDEST PERSON	Model	IncomeGroup	ESTIMATED BIRTHDATE
1	95	Bike Wash	Low	1925
2	82	Racing Socks	Moderate	1938
3	86	Water Bottle	Moderate	1934
4	86	Half-Finger Gloves	Moderate	1934
5	75	Road-550-W	High	1945
6	82	Mountain-200	Low	1938
7	74	Touring-2000	Low	1946
8	86	HL Road Tire	Low	1934
9	82	Road Bottle Cage	High	1938
10	97	Road Tire Tube	Moderate	1923

Ln 209, Col 1 (346 selected) Spaces: 4 UTF-8 LF SQL MSSQL 46 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Customer')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  [
2      "Customer": [
3          {
4              "OLDEST PERSON": 95,
5              "Model": "Bike Wash",
6              "IncomeGroup": "Low",
7              "ESTIMATED BIRTHDATE": 1925
8          },
9          {
10             "OLDEST PERSON": 82,
11             "Model": "Racing Socks",
12             "IncomeGroup": "Moderate",
13             "ESTIMATED BIRTHDATE": 1938
14         },
15         {
16             "OLDEST PERSON": 86,
17             "Model": "Water Bottle",
18             "IncomeGroup": "Moderate",
19             "ESTIMATED BIRTHDATE": 1934
20         },
21         {
22             "OLDEST PERSON": 86,
23             "Model": "Half-Finger Gloves",
24             "IncomeGroup": "Moderate",
25             "ESTIMATED BIRTHDATE": 1934
26         },

```

Proposition 14

[Diagram of Database](#)

Relational Crossjoin subsystem in Northwinds2019TSQLV5

RelationalCrossJoin.S1



RelationalCrossJoin.S2

Problem Statement

Find every combination of letters a-d and numbers 1-3

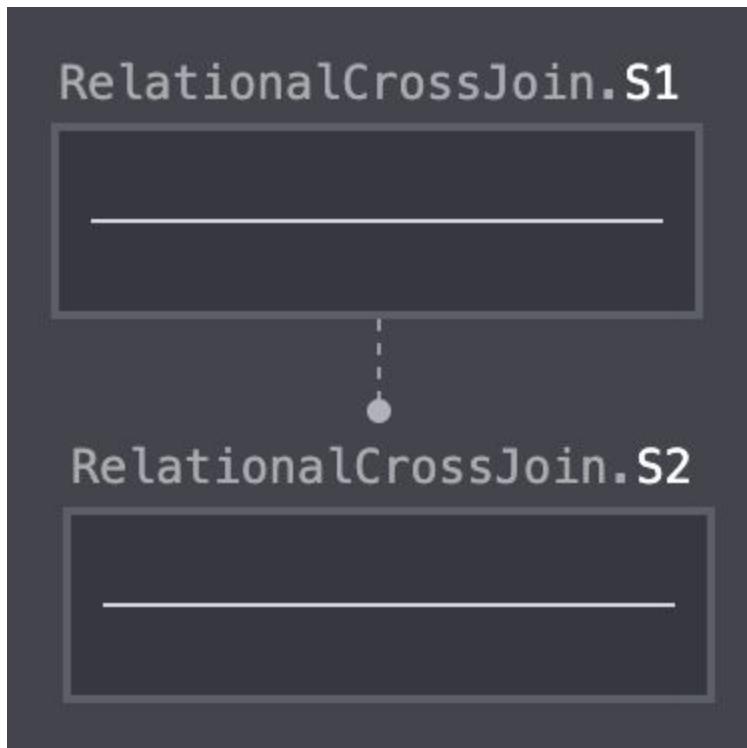
Database

Northwinds2019TSQLV5

DATABASE PROPOSITIONS AND SOLUTIONS

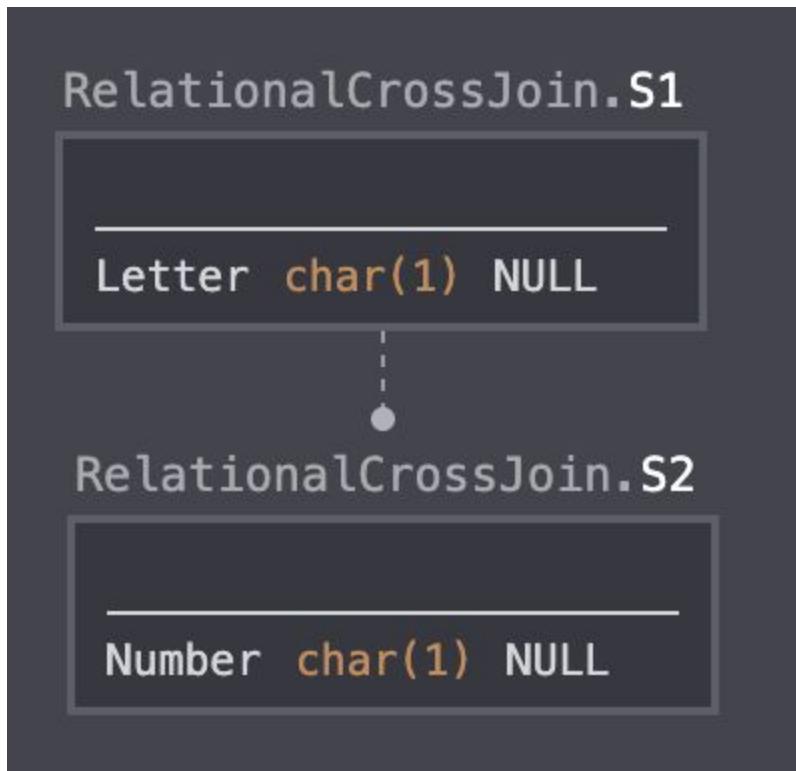
Diagrams of Tables

Key View



DATABASE PROPOSITIONS AND SOLUTIONS

Standard View



Attributes

Table Name	Column Name
RelationalCrossJoin	S1
RelationalCrossJoin	S2

Problem-solving Query

```
USE Northwinds2019TSQLV5
GO
```

```
SELECT *
FROM RelationalCrossJoin.S1
CROSS JOIN RelationalCrossJoin.S2
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	Letter	Number
1	a	1
2	b	1
3	c	1
4	d	1
5	a	2
6	b	2
7	c	2
8	d	2
9	a	3
10	b	3

Ln 52, Col 1 (102 selected) Spaces: 4 UTF-8 LF SQL MSSQL 12 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Combinations')
```

DATABASE PROPOSITIONS AND SOLUTIONS

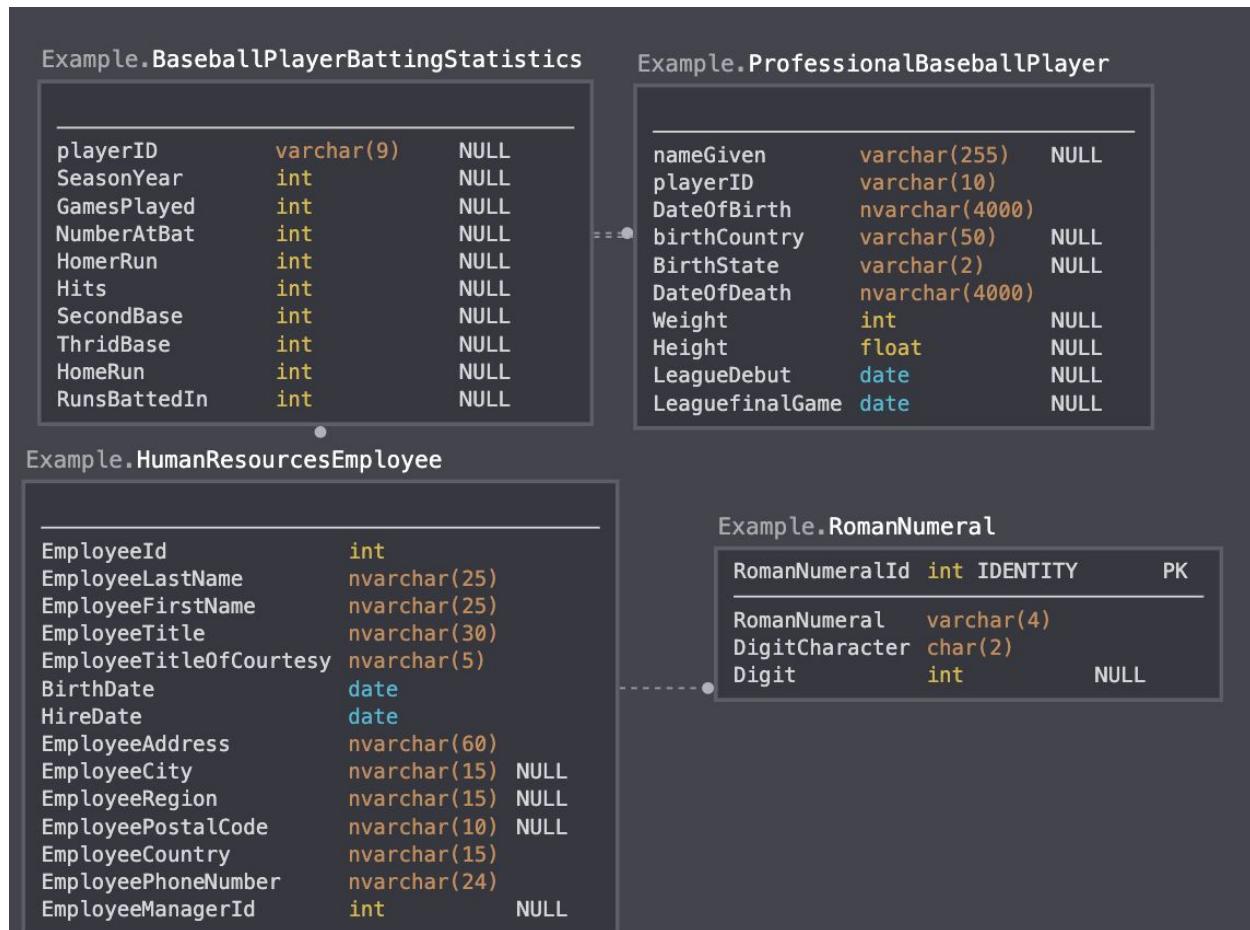
```
1  {
2      "Combinations": [
3          {
4              "Letter": "a",
5              "Number": "1"
6          },
7          {
8              "Letter": "b",
9              "Number": "1"
10         },
11        {
12            "Letter": "c",
13            "Number": "1"
14        },
15        {
16            "Letter": "d",
17            "Number": "1"
18        },
19        {
20            "Letter": "a",
21            "Number": "2"
22        },
23        {
24            "Letter": "b",
25            "Number": "2"
26        },
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 15

Diagram of Database

Example subsystem in Northwinds2019TSQLV5



Problem Statement

Find the average number of games played by nationality, and associate each with a number

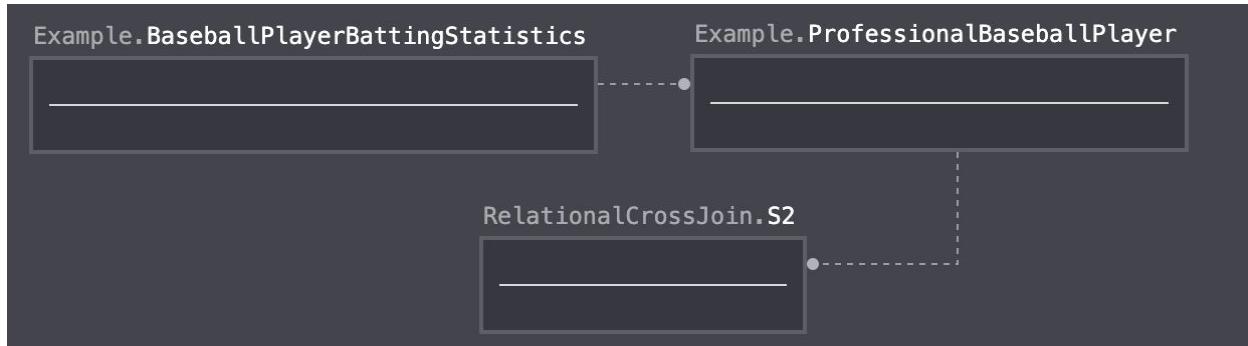
Database

Northwinds2019TSQLV5

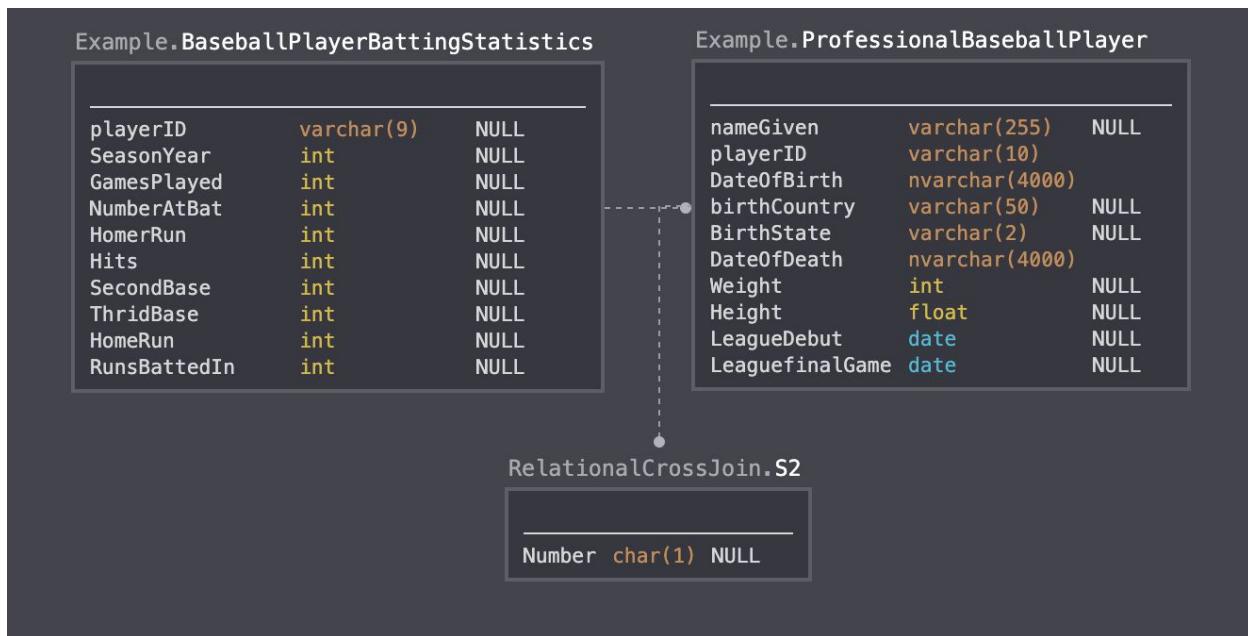
DATABASE PROPOSITIONS AND SOLUTIONS

Diagrams of Tables

Key View



Standard View



Attributes

Table Name	Column Name
Example.BaseballPlayerBattingStatistics	birthCountry
Example.ProfessionalBaseballPlayer	nationality
RelationalCrossJoin.S2	Number

Problem-solving Query

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
  
```

Prepared by: Ibrahim Suhail

Date Prepared: 3/21/2020

C:\Users\Ibrahim Suhail\Desktop\Spring 2020\CS331\Project 1\G12-5 - Ibrahim Suhail.docx

Page 72 of 88

DATABASE PROPOSITIONS AND SOLUTIONS

```
GO
create function [dbo].[nationality](
    @p NVARCHAR(10)
)
RETURNS
NVARCHAR(10)

begin
    DECLARE @res NVARCHAR(10)
    select @res = case
        when @p = 'D.R.' then 'Dominican'
        when @p = 'USA' then 'American'
        else 'unknown'
    end;
    return @res
end;
GO

USE Northwinds2019TSQLV5
GO

SELECT TOP 40 avg(a.GamesPlayed) AS [AVG_GAMES_PLAYED]
    ,birthCountry
    ,example.nationality(b.birthCountry) AS [nationality]
    ,s.Number
FROM Example.BaseballPlayerBattingStatistics AS a
JOIN Example.ProfessionalBaseballPlayer AS b ON a.playerID = b.playerID
LEFT OUTER JOIN RelationalCrossJoin.S2 AS s ON birthCountry >= s.Number
GROUP BY birthCountry
    ,s.Number
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

Results					Messages
	AVG_GAMES_PLAYED	birthCountry	nationality	Number	
1	2	Lithuania	Lithuanian	1	
2	46	Germany	unknown	3	
3	13	Saudi Arabia	unknown	2	
4	14	Sweden	unknown	1	
5	7	Czech Republic	unknown	1	
6	67	Panama	Panaman	1	
7	77	Colombia	Colombian	1	
8	23	Aruba	Aruban	3	
9	33	Italy	unknown	2	
10	58	Cuba	Cuban	1	

Ln 138, Col 1 (410 selected) Spaces: 4 UTF-8 LF SQL MSSQL 40 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Player')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  {
2      "Player": [
3          {
4              "AVG_GAMES_PLAYED": 2,
5              "birthCountry": "Lithuania",
6              "nationality": "Lithuanian",
7              "Number": "1"
8          },
9          {
10             "AVG_GAMES_PLAYED": 46,
11             "birthCountry": "Germany",
12             "nationality": "unknown",
13             "Number": "3"
14         },
15         {
16             "AVG_GAMES_PLAYED": 13,
17             "birthCountry": "Saudi Arabia",
18             "nationality": "unknown",
19             "Number": "2"
20         },
21         {
22             "AVG_GAMES_PLAYED": 14,
23             "birthCountry": "Sweden",
24             "nationality": "unknown",
25             "Number": "1"
26         },

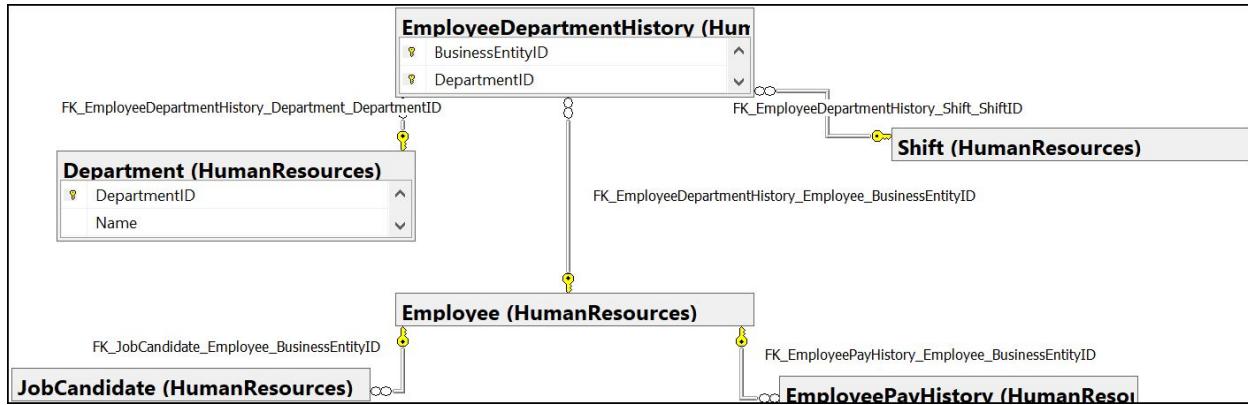
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 16

Diagram of Database

HumanResources subsystem in adventureworks2014



Problem Statement

Find the department name of every employee

Database

adventureworks2014

Attributes

Table Name	Column Name
EmployeeDepartmentHistory	BusinessEntityID DepartmentID
Department	Name

Order by

Table Name	Column Name	Sort Order
EmployeeDepartmentHistory	BusinessEntityID	ASC

Problem-solving Query

USE adventureworks2014

GO

```

SELECT edh.BusinessEntityID
      ,edh.DepartmentID
      ,d.Name
FROM HumanResources.EmployeeDepartmentHistory AS edh
JOIN HumanResources.Department AS d ON edh.DepartmentID = d.DepartmentID
ORDER BY BusinessEntityID
  
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

Results				Messages	
	BusinessEntityID	DepartmentID	Name		
1	1	16	Executive		
2	2	1	Engineering		
3	3	1	Engineering		
4	4	1	Engineering		
5	4	2	Tool Design		
6	5	1	Engineering		
7	6	1	Engineering		
8	7	6	Research a...		
9	8	6	Research a...		
...	9	6	Research a...		
...	10	6	Research a...		

Ln 1, Col 1 (317 selected) Spaces: 4 UTF-8 LF SQL MSSQL 296 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Department')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  {
2      "Department": [
3          {
4              "BusinessEntityID": 1,
5              "DepartmentID": 16,
6              "Name": "Executive"
7          },
8          {
9              "BusinessEntityID": 2,
10             "DepartmentID": 1,
11             "Name": "Engineering"
12         },
13         {
14             "BusinessEntityID": 3,
15             "DepartmentID": 1,
16             "Name": "Engineering"
17         },
18         {
19             "BusinessEntityID": 4,
20             "DepartmentID": 1,
21             "Name": "Engineering"
22         },
23         {
24             "BusinessEntityID": 4,
25             "DepartmentID": 2,
26             "Name": "Tool Design"
27         },

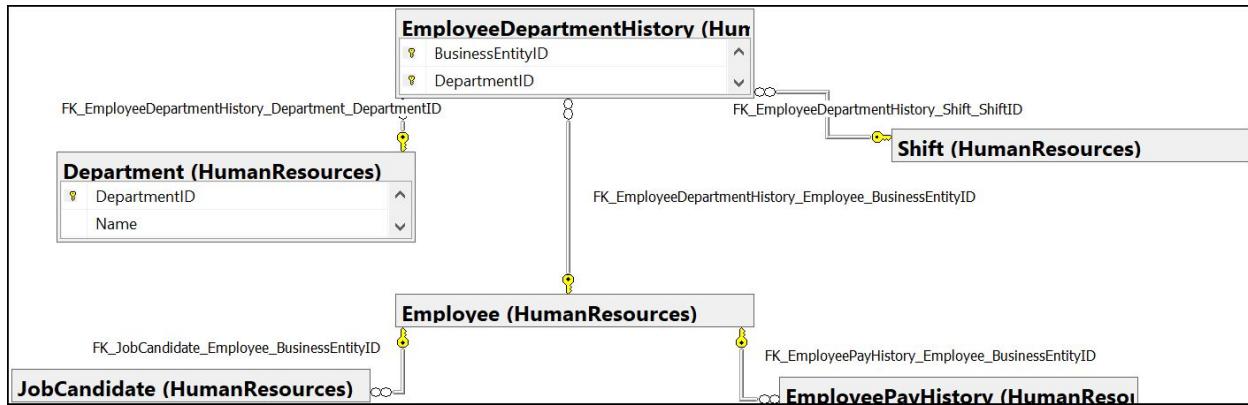
```

Proposition 17

Diagram of Database

Human Resources subsystem in adventureworks2014

DATABASE PROPOSITIONS AND SOLUTIONS



Problem Statement

Find all male day employees and their job titles

Database

adventureworks2014

Attributes

Table Name	Column Name
Employee	BusinessEntityID Gender JobTitle
EmployeeDepartmentHistory	
Shift	Name

Problem-solving Query

```
USE adventureworks2014
```

```
GO
```

```

SELECT e.BusinessEntityID
      ,e.Gender
      ,s.Name
      ,e.JobTitle
FROM HumanResources.Employee AS e
JOIN HumanResources.EmployeeDepartmentHistory AS dh ON e.BusinessEntityID =
dh.BusinessEntityID
JOIN HumanResources.Shift AS s ON dh.ShiftID = s.ShiftID
WHERE Gender = 'm'
      AND Name = 'day'
```

DATABASE PROPOSITIONS AND SOLUTIONS

Sample Relational Output

	BusinessEntityID	Gender	Name	JobTitle	
1	1	M	Day	Chief Executive Officer	
2	3	M	Day	Engineering Manager	
3	4	M	Day	Senior Tool Designer	
4	4	M	Day	Senior Tool Designer	
5	6	M	Day	Design Engineer	
6	7	M	Day	Research and Development ...	
7	10	M	Day	Research and Development ...	
8	11	M	Day	Senior Tool Designer	
9	12	M	Day	Tool Designer	
...	14	M	Day	Senior Design Engineer	
...	16	M	Day	Marketing Manager	
	16	M	Day	Marketing Manager	

Ln 26, Col 21 Tab Size: 4 UTF-8 LF SQL MSSQL 123 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Employee')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  {
2      "Employee": [
3          {
4              "BusinessEntityID": 1,
5              "Gender": "M",
6              "Name": "Day",
7              "JobTitle": "Chief Executive Officer"
8          },
9          {
10             "BusinessEntityID": 3,
11             "Gender": "M",
12             "Name": "Day",
13             "JobTitle": "Engineering Manager"
14         },
15         {
16             "BusinessEntityID": 4,
17             "Gender": "M",
18             "Name": "Day",
19             "JobTitle": "Senior Tool Designer"
20         },

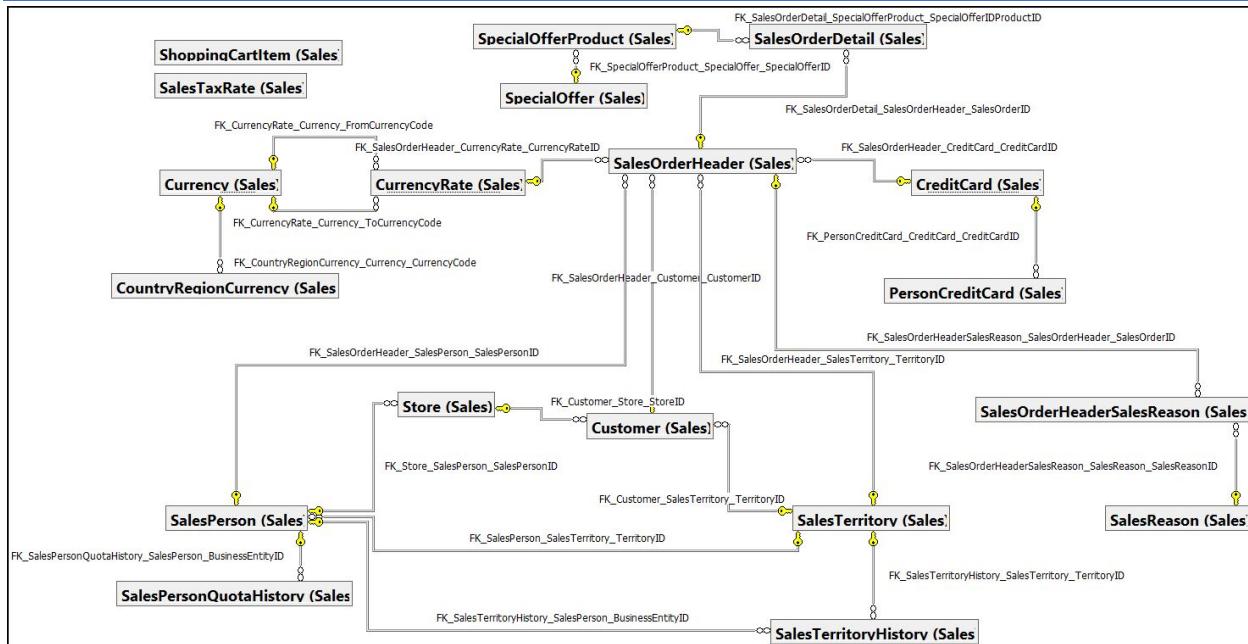
```

Proposition 18

[Diagram of Database](#)

Sales subsystem in Adventureworks2014

DATABASE PROPOSITIONS AND SOLUTIONS



Problem Statement

Write a query that returns the percent of total revenue accounted for each category of products in 2016.

Database

adventureworks2014

Attributes

Table Name	Column Name
BusinessEntityAddress	BusinessEntityID AddressID AddressTypeID name
AddressType	name

Problem-solving Query

```
USE adventureworks2014
GO
```

```
SELECT TOP 100 e.BusinessEntityID
      ,e.AddressID
      ,e.AddressTypeID
      ,a.name
  FROM Person.BusinessEntityAddress AS e
```

Prepared by: Ibrahim Suhail

Date Prepared: 3/21/2020

C:\Users\Ibrahim Suhail\Desktop\Spring 2020\CS331\Project 1\G12-5 - Ibrahim Suhail.docx

Page 82 of 88

DATABASE PROPOSITIONS AND SOLUTIONS

```
JOIN Person.AddressType AS a ON a.AddressTypeID = e.AddressTypeID  
ORDER BY e.AddressID
```

Sample Relational Output

	BusinessEntityID	AddressID	AddressTypeID	name
1	12	1	2	Home
2	123	2	2	Home
3	285	3	2	Home
4	251	4	2	Home
5	124	5	2	Home
6	210	6	2	Home
7	88	7	2	Home
8	59	8	2	Home
9	1470	9	5	Shipping
10	212	10	2	Home

Ln 23, Col 1 (235 selected) Spaces: 4 UTF-8 LF SQL MSSQL 100 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Address')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  [
2      "Address": [
3          {
4              "BusinessEntityID": 12,
5              "AddressID": 1,
6              "AddressTypeID": 2,
7              "name": "Home"
8          },
9          {
10             "BusinessEntityID": 123,
11             "AddressID": 2,
12             "AddressTypeID": 2,
13             "name": "Home"
14         },
15         {
16             "BusinessEntityID": 285,
17             "AddressID": 3,
18             "AddressTypeID": 2,
19             "name": "Home"
20         },
21         {
22             "BusinessEntityID": 251,
23             "AddressID": 4,
24             "AddressTypeID": 2,
25             "name": "Home"
26         },

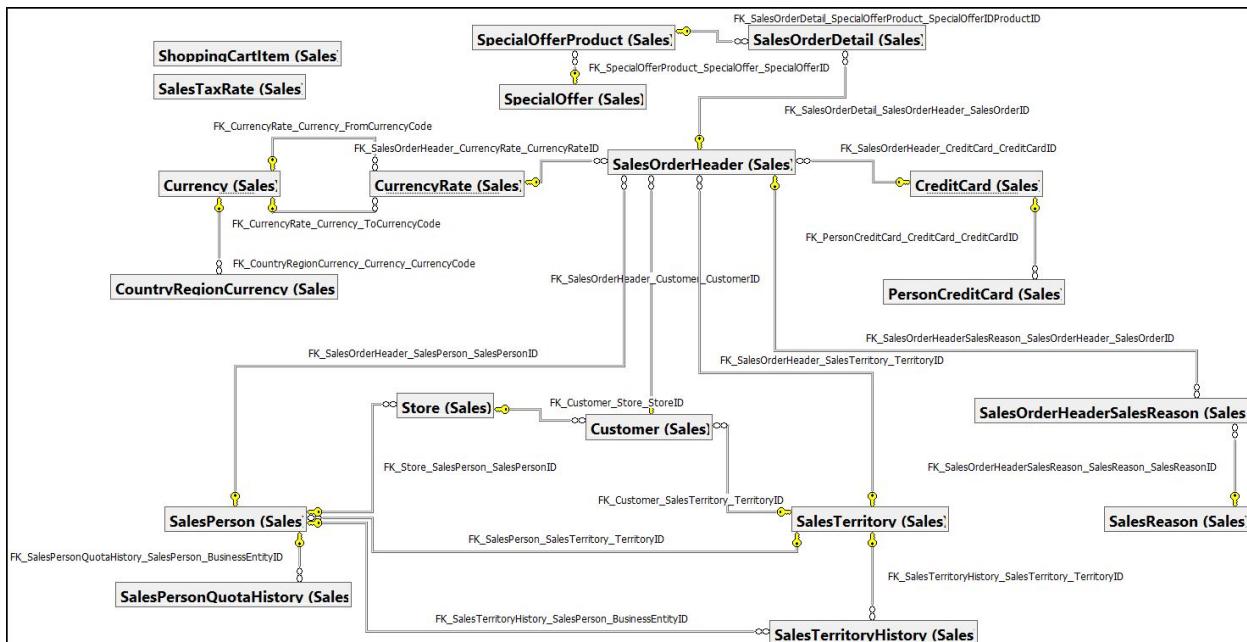
```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 19

Diagram of Database

Sales subsystem in adventureworks2014



Problem Statement

Given 100 phone numbers, find the phone number name

Database

adventureworks2014

Attributes

Table Name	Column Name
Product	ProductID Discontinued ProductName
OrderDetail	ProductID OrderID
Order	OrderID OrderDate

Problem-solving Query

USE adventureworks2014

Prepared by: Ibrahim Suhail

C:\Users\Ibrahim Suhail\Desktop\Spring 2020\CS331\Project 1\G12-5 - Ibrahim Suhail.docx

Page 85 of 88

Date Prepared: 3/21/2020

DATABASE PROPOSITIONS AND SOLUTIONS

```
GO
SELECT TOP 100 PhoneNumber
    ,name

FROM Person.PersonPhone AS ppp
JOIN Person.PhoneNumberType AS pnt ON ppp.PhoneNumberTypeID =
pnt.PhoneNumberTypeID
ORDER BY PhoneNumber
```

Sample Relational Output

	PhoneNumber	name
1	1 (11) 500 555-0110	Work
2	1 (11) 500 555-0110	Cell
3	1 (11) 500 555-0110	Home
4	1 (11) 500 555-0110	Cell
5	1 (11) 500 555-0110	Home
6	1 (11) 500 555-0110	Home
7	1 (11) 500 555-0110	Home
8	1 (11) 500 555-0110	Cell
9	1 (11) 500 555-0110	Home
10	1 (11) 500 555-0110	Cell

Ln 37, Col 21 Spaces: 4 UTF-8 LF SQL MSSQL 100 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Phone Number')
```

DATABASE PROPOSITIONS AND SOLUTIONS

```
1  [
2      "Phone Number": [
3          {
4              "PhoneNumber": "1 (11) 500 555-0110",
5              "name": "Work"
6          },
7          {
8              "PhoneNumber": "1 (11) 500 555-0110",
9              "name": "Cell"
10         },
11         {
12             "PhoneNumber": "1 (11) 500 555-0110",
13             "name": "Home"
14         },
15         {
16             "PhoneNumber": "1 (11) 500 555-0110",
17             "name": "Cell"
18         },
19         {
20             "PhoneNumber": "1 (11) 500 555-0110",
21             "name": "Home"
22         },
23         {
24             "PhoneNumber": "1 (11) 500 555-0110",
25             "name": "Home"
26         },

```

DATABASE PROPOSITIONS AND SOLUTIONS

Proposition 20

Diagram of Database

Stats subsystem in Northwinds2019TSQLV5



Problem Statement

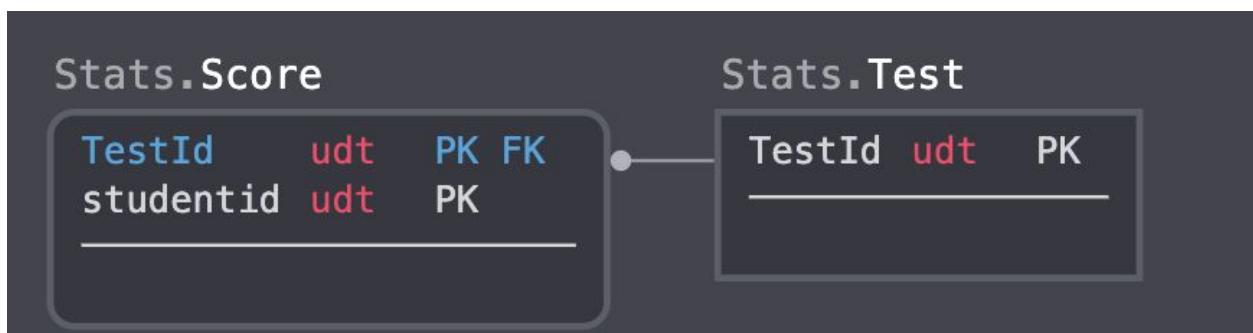
Find the average, lowest, and highest score per test

Database

Northwinds2019TSQLV5

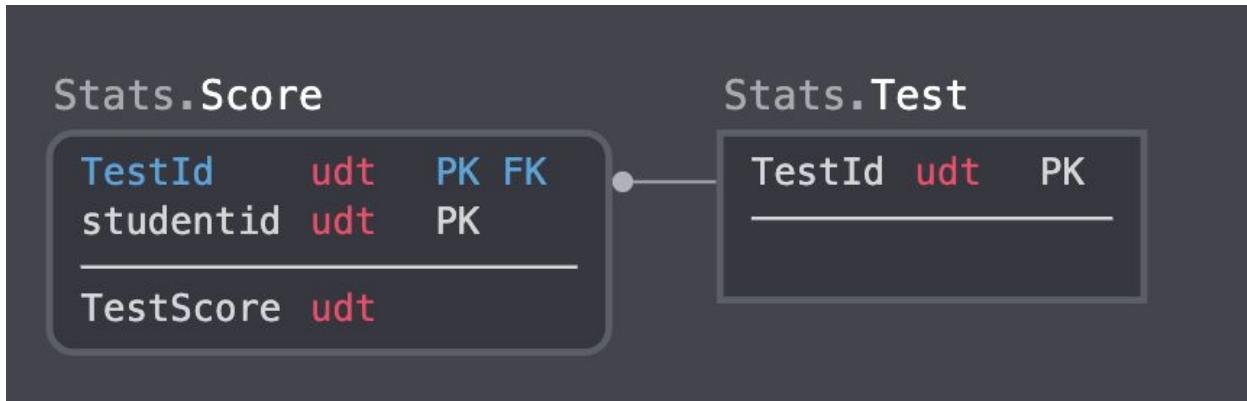
Diagrams of Tables

Key View



DATABASE PROPOSITIONS AND SOLUTIONS

Standard View



Attributes

Table Name	Column Name
Score	EmployeeID EmployeeFirstName EmployeeLastName
Test	OrderID EmployeeID

Problem-solving Query

```

USE Northwinds2019TSQLV5
GO
SELECT s.testid
    ,avg(s.TestScore) AS [MEAN SCORE]
    ,MIN(S.TestScore) AS [LOWEST SCORE]
    ,MAX(TestScore) AS [HIGHEST SCORE]
FROM Stats.Score AS s
JOIN Stats.Test AS t ON s.TestId = t.TestId
GROUP BY S.TestId
  
```

DATABASE PROPOSITIONS AND SOLUTIONS

	testid	MEAN SCORE	LOWEST SCORE	HIGHEST SCORE
1	Test ABC	72	50	95
2	Test XYZ	74	50	95

Ln 46, Col 22 Spaces: 4 UTF-8 LF SQL MSSQL 2 rows

Sample JSON Output

Remove semicolon from last line of query, and add:

```
for json PATH, root('Test')
```

```
1  {
2      "Test": [
3          {
4              "testid": "Test ABC",
5              "MEAN SCORE": 72,
6              "LOWEST SCORE": 50,
7              "HIGHEST SCORE": 95
8          },
9          {
10             "testid": "Test XYZ",
11             "MEAN SCORE": 74,
12             "LOWEST SCORE": 50,
13             "HIGHEST SCORE": 95
14         }
15     ]
16 }
```