

```
1  --1 . List of all customers
2  SELECT * FROM Sales.Customer;
3
4
5
6  --2.List of all customers where company name ends with 'N':
7  SELECT *
8  FROM Sales.Store s
9  JOIN Sales.Customer c ON s.BusinessEntityID = c.StoreID
10 WHERE s.Name LIKE '%N';
11
12
13
14 --3-List of all customers who live in Berlin or London:
15 SELECT c.*
16 FROM Sales.Customer c
17 JOIN Person.BusinessEntityAddress bea ON c.CustomerID =
    bea.BusinessEntityID
18 JOIN Person.Address a ON bea.AddressID = a.AddressID
19 WHERE a.City IN ('Berlin', 'London');
20
21
22
23 --4-List of all customers who live in UK or USA
24 SELECT c.*
25 FROM Sales.Customer c
26 JOIN Person.BusinessEntityAddress bea ON c.CustomerID =
    bea.BusinessEntityID
27 JOIN Person.Address a ON bea.AddressID = a.AddressID
28 JOIN Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
29 WHERE sp.CountryRegionCode IN ('UK', 'US');
30
31
32
33 --5-List of all products sorted by product name
34 SELECT Name
35 FROM Production.Product
36 ORDER BY Name;
37
38
39
40 --6-List of all products where product name starts with an A
41 SELECT ProductID, Name, ProductNumber, MakeFlag, FinishedGoodsFlag,
    Color, SafetyStockLevel, ReorderPoint, StandardCost, ListPrice,
    Size, SizeUnitMeasureCode, WeightUnitMeasureCode, Weight,
    DaysToManufacture, ProductLine, Class, Style, ProductSubcategoryID,
    ProductModelID, SellStartDate, SellEndDate, DiscontinuedDate,
    rowguid, ModifiedDate
42 FROM Production.Product
43 WHERE Name LIKE 'A%'
44 ORDER BY Name;
45
46
```

```
47
48 --7-List of customers who ever placed an order
49 SELECT DISTINCT c.*
50 FROM Sales.Customer c
51 JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID;
52
53
54
55 --8-List of customers who live in london and have bought chai
56 SELECT DISTINCT c.*
57 FROM Sales.Customer c
58 JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID
59 JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
60 JOIN Production.Product p ON sod.ProductID = p.ProductID
61 JOIN Person.BusinessEntityAddress bea ON soh.ShipToAddressID =
    bea.BusinessEntityID
62 JOIN Person.Address a ON bea.AddressID = a.AddressID
63 WHERE a.City = 'London' AND p.Name = 'Chai';
64
65
66
67 --9-List of customer who never placed an order
68 SELECT * FROM Sales.Customer c
69 WHERE NOT EXISTS (
70     SELECT 1
71     FROM Sales.SalesOrderHeader soh
72     WHERE c.CustomerID = soh.CustomerID
73 );
74
75
76
77 --10-List of customers who ordered a Tofu
78 SELECT DISTINCT c.*
79 FROM Sales.Customer c
80 JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID
81 JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
82 JOIN Production.Product p ON sod.ProductID = p.ProductID
83 WHERE p.Name = 'Tofu';
84
85
86
87 --11-Details of first order of the system
88 SELECT TOP 1 *
89 FROM Sales.SalesOrderHeader
90 ORDER BY OrderDate ASC;
91
92
93
94 --12-Find the details of most expensive order date
95 WITH OrderTotals AS (
96     SELECT
97         SalesOrderID,
98         SUM(LineTotal) AS TotalOrderCost
```

```
99      FROM
100      Sales.SalesOrderDetail
101      GROUP BY
102      SalesOrderID
103  )
104  SELECT
105      soh.SalesOrderID,
106      soh.OrderDate,
107      soh.DueDate,
108      soh.ShipDate,
109      soh.Status,
110      soh.OnlineOrderFlag,
111      soh.SalesOrderNumber,
112      soh.PurchaseOrderNumber,
113      soh.AccountNumber,
114      soh.CustomerID,
115      soh.SalesPersonID,
116      soh.TerritoryID,
117      soh.BillToAddressID,
118      soh.ShipToAddressID,
119      soh.ShipMethodID,
120      soh.CreditCardID,
121      soh.CreditCardApprovalCode,
122      soh.CurrencyRateID,
123      soh.SubTotal,
124      soh.TaxAmt,
125      soh.Freight,
126      soh.TotalDue,
127      soh.Comment,
128      soh.rowguid,
129      soh.ModifiedDate,
130      ot.TotalOrderCost
131  FROM
132      Sales.SalesOrderHeader soh
133  JOIN
134      OrderTotals ot ON soh.SalesOrderID = ot.SalesOrderID
135  ORDER BY
136      ot.TotalOrderCost DESC
137
138
139
140  --13- for each order get theorderid and average quantity of items in that order
141  SELECT
142      SalesOrderID,
143      AVG(OrderQty) AS AverageQuantity
144  FROM
145      Sales.SalesOrderDetail
146  GROUP BY
147      SalesOrderID;
148
149
150
```

```
151 --14-For each order get the OrderID,minimum quantity and maximum quantity for that order
152 SELECT
153     SalesOrderID,
154     MIN(OrderQty) AS MinimumQuantity,
155     MAX(OrderQty) AS MaximumQuantity
156 FROM
157     Sales.SalesOrderDetail
158 GROUP BY
159     SalesOrderID;
160
161
162
163 --15-Get a list of all managers and total no of employees who report to them.
164 USE AdventureWork;
165
166 SELECT
167     E.BusinessEntityID AS ManagerID,
168     P.FirstName + ' ' + P.LastName AS ManagerName,
169     COUNT(RE.BusinessEntityID) AS TotalEmployees
170 FROM
171     HumanResources.Employee AS E
172 INNER JOIN
173     Person.Person AS P ON E.BusinessEntityID = P.BusinessEntityID
174 LEFT JOIN
175     HumanResources.Employee AS RE ON RE.OrganizationNode.GetAncestor(1) = E.OrganizationNode
176 GROUP BY
177     E.BusinessEntityID, P.FirstName, P.LastName
178 ORDER BY
179     TotalEmployees DESC;
180
181
182
183 --16-Get the orderID and total quantity for each order that has a total quantity of greater than 300.
184 SELECT
185     SalesOrderID,
186     SUM(OrderQty) AS TotalQuantity
187 FROM
188     Sales.SalesOrderDetail
189 GROUP BY
190     SalesOrderID
191 HAVING
192     SUM(OrderQty) > 300;
193
194 --17-List of all the orders placed on after 1996/12/31.
195 SELECT *
196 FROM Sales.SalesOrderHeader
197 WHERE OrderDate >= '1996-12-31';
198
199 --18-List of all orders shipped to Canada.
```

```
200 SELECT
201     soh.SalesOrderID,
202     soh.OrderDate,
203     soh.ShipDate,
204     soh.Status,
205     soh.TotalDue,
206     a.AddressLine1,
207     a.AddressLine2,
208     a.City,
209     sp.Name AS StateProvince,
210     cr.Name AS CountryRegion
211 FROM
212     Sales.SalesOrderHeader soh
213 JOIN
214     Person.Address a ON soh.ShipToAddressID = a.AddressID
215 JOIN
216     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
217 JOIN
218     Person.CountryRegion cr ON sp.CountryRegionCode =
219                                     cr.CountryRegionCode
220 WHERE
221     cr.Name = 'Canada';
222
223
224 --19-List of all orders with order total > 200 .
225 SELECT *
226 FROM Sales.SalesOrderHeader
227 WHERE TotalDue > 200;
228
229 --20-List of all countries and sales made in each country .
230 SELECT
231     cr.Name AS Country,
232     SUM(soh.TotalDue) AS TotalSales
233 FROM
234     Sales.SalesOrderHeader soh
235 JOIN
236     Person.Address a ON soh.ShipToAddressID = a.AddressID
237 JOIN
238     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
239 JOIN
240     Person.CountryRegion cr ON sp.CountryRegionCode =
241                                     cr.CountryRegionCode
242 GROUP BY
243     cr.Name
244 ORDER BY
245     TotalSales DESC;
246
247
248 --21-List of customers Contact name and numbers of orders they placed .
249 SELECT
250     p.FirstName + ' ' + p.LastName AS ContactName,
```

```
251     COUNT(soh.SalesOrderID) AS NumberOfOrders
252 FROM
253     Sales.SalesOrderHeader soh
254 JOIN
255     Sales.Customer c ON soh.CustomerID = c.CustomerID
256 JOIN
257     Person.Person p ON c.PersonID = p.BusinessEntityID
258 GROUP BY
259     p.FirstName, p.LastName
260 ORDER BY
261     NumberOfOrders DESC;
262
263
264
265 --22-List of customers contactnames who have placed more than 3 orders.
266 USE AdventureWork;
267
268 SELECT
269     P.FirstName + ' ' + P.LastName AS ContactName,
270     COUNT(SOH.SalesOrderID) AS NumberOfOrders
271 FROM
272     Sales.Customer AS C
273 INNER JOIN
274     Sales.SalesOrderHeader AS SOH ON C.CustomerID = SOH.CustomerID
275 INNER JOIN
276     Person.Person AS P ON C.PersonID = P.BusinessEntityID
277 GROUP BY
278     P.FirstName, P.LastName
279 HAVING
280     COUNT(SOH.SalesOrderID) > 3
281 ORDER BY
282     NumberOfOrders DESC;
283
284
285
286 --23-List of discontinued products which were ordered between 1/1/1997 and 1/1/1998
287 USE AdventureWork;
288
289 SELECT
290     P.ProductID,
291     P.Name AS ProductName,
292     P.DiscontinuedDate
293 FROM
294     Sales.SalesOrderDetail AS SOD
295 INNER JOIN
296     Sales.SalesOrderHeader AS SOH ON SOD.SalesOrderID =
297     SOH.SalesOrderID
298 INNER JOIN
299     Production.Product AS P ON SOD.ProductID = P.ProductID
300 WHERE
301     SOH.OrderDate BETWEEN '1997-01-01' AND '1998-01-01'
302     AND P.DiscontinuedDate IS NOT NULL
```

```
302     AND P.DiscontinuedDate <= '1998-01-01'
303 GROUP BY
304     P.ProductID, P.Name, P.DiscontinuedDate
305 ORDER BY
306     P.ProductID;
307
308
309
310 --24-List of employees firstname,lastname,supervisor Firstname,Lastname
311 USE AdventureWork;
312
313 SELECT
314     E1.FirstName AS EmployeeFirstName,
315     E1.LastName AS EmployeeLastName,
316     E2.FirstName AS SupervisorFirstName,
317     E2.LastName AS SupervisorLastName
318 FROM
319     HumanResources.Employee AS Emp1
320 INNER JOIN
321     Person.Person AS E1 ON Emp1.BusinessEntityID = E1.BusinessEntityID
322 LEFT JOIN
323     HumanResources.Employee AS Emp2 ON
324         Emp1.OrganizationNode.GetAncestor(1) = Emp2.OrganizationNode
325     Person.Person AS E2 ON Emp2.BusinessEntityID = E2.BusinessEntityID
326 ORDER BY
327     EmployeeLastName, EmployeeFirstName;
328
329
330
331 --25-List of employees ID and total sale conducted by employee.
332 USE AdventureWork;
333
334 SELECT
335     E.BusinessEntityID AS EmployeeID,
336     SUM(SOH.TotalDue) AS TotalSales
337 FROM
338     Sales.SalesOrderHeader AS SOH
339 INNER JOIN
340     Sales.SalesPerson AS SP ON SOH.SalesPersonID = SP.BusinessEntityID
341 INNER JOIN
342     HumanResources.Employee AS E ON SP.BusinessEntityID =
343         E.BusinessEntityID
344 GROUP BY
345     E.BusinessEntityID
346 ORDER BY
347     TotalSales DESC;
348
349
350 --26-List of employees whose Firstname contains character a.
351 USE AdventureWork;
352 SELECT FirstName, LastName
```

```
353 FROM HumanResources.Employee AS e
354 JOIN Person.Person AS p ON e.BusinessEntityID = p.BusinessEntityID
355 WHERE p.FirstName LIKE '%a%';
356
357
358
359 --27-List of managers who have more than four people reporting to them.
360 USE AdventureWork;
361
362 SELECT
363     ManagerPerson.FirstName,
364     ManagerPerson.LastName,
365     COUNT(Employee.BusinessEntityID) AS ReportCount
366 FROM
367     HumanResources.Employee AS Employee
368 JOIN
369     HumanResources.Employee AS Manager ON
370         Employee.OrganizationNode.GetAncestor(1) =
371         Manager.OrganizationNode
372 JOIN
373     Person.Person AS ManagerPerson ON Manager.BusinessEntityID =
374         ManagerPerson.BusinessEntityID
375 GROUP BY
376     ManagerPerson.FirstName,
377     ManagerPerson.LastName
378 HAVING
379     COUNT(Employee.BusinessEntityID) > 4;
380
381 --28-List of orders and ProductNames.
382 USE AdventureWork;
383 SELECT
384     soh.SalesOrderID,
385     soh.OrderDate,
386     p.Name AS ProductName
387 FROM
388     Sales.SalesOrderHeader soh
389 JOIN
390     Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
391 JOIN
392     Production.Product p ON sod.ProductID = p.ProductID;
393
394
395 --29-List of orders placed by best customer.
396
397 --identify the best customer.
398 USE AdventureWork;
399 WITH CustomerTotal AS (
400     SELECT
401         soh.CustomerID,
402         SUM(soh.TotalDue) AS TotalSpent
```



```
403     FROM
404         Sales.SalesOrderHeader soh
405     GROUP BY
406         soh.CustomerID
407 )
408 SELECT TOP 1
409     CustomerID,
410     TotalSpent
411 FROM
412     CustomerTotal
413 ORDER BY
414     TotalSpent DESC;
415
416 --order placed by the best customer.
417 USE AdventureWork;
418 SELECT
419     soh.SalesOrderID,
420     soh.OrderDate,
421     soh.TotalDue
422 FROM
423     Sales.SalesOrderHeader soh
424 WHERE
425     soh.CustomerID = (
426         SELECT TOP 1
427             CustomerID
428         FROM
429             (
430                 SELECT
431                     soh.CustomerID,
432                     SUM(soh.TotalDue) AS TotalSpent
433                 FROM
434                     Sales.SalesOrderHeader soh
435                 GROUP BY
436                     soh.CustomerID
437             ) AS CustomerTotal
438         ORDER BY
439             TotalSpent DESC
440     );
441
442
443
444 --30-List of orders placed by customer who do not have a fax number.
445 USE AdventureWork;
446
447 SELECT
448     soh.SalesOrderID,
449     soh.OrderDate,
450     soh.TotalDue,
451     c.CustomerID,
452     p.FirstName,
453     p.LastName
454 FROM
455     Sales.SalesOrderHeader soh
```

```
456 JOIN
457     Sales.Customer c ON soh.CustomerID = c.CustomerID
458 JOIN
459     Person.Person p ON c.PersonID = p.BusinessEntityID
460 LEFT JOIN
461     Person.PersonPhone pp ON p.BusinessEntityID = pp.BusinessEntityID
462     AND pp.PhoneNumberTypeID = (SELECT PhoneNumberTypeID FROM
463     Person.PhoneNumberType WHERE Name = 'Fax')
464 WHERE
465     pp.PhoneNumber IS NULL;
466
467 --31-List of postal codes where the product tofu was shipped.
468 USE AdventureWork;
469
470 SELECT DISTINCT
471     a.PostalCode
472 FROM
473     Sales.SalesOrderDetail sod
474 JOIN
475     Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
476 JOIN
477     Production.Product p ON sod.ProductID = p.ProductID
478 JOIN
479     Person.Address a ON soh.ShipToAddressID = a.AddressID
480 WHERE
481     p.Name = 'Tofu';
482
483
484
485 --32-List of product Names that were shipped to France.
486 USE AdventureWork;
487
488 SELECT DISTINCT
489     p.Name AS ProductName
490 FROM
491     Sales.SalesOrderDetail sod
492 JOIN
493     Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
494 JOIN
495     Production.Product p ON sod.ProductID = p.ProductID
496 JOIN
497     Person.Address a ON soh.ShipToAddressID = a.AddressID
498 JOIN
499     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
500 JOIN
501     Person.CountryRegion cr ON sp.CountryRegionCode =
502     cr.CountryRegionCode
503 WHERE
504     cr.Name = 'France';
505
```

```
506
507 --33-List of ProductNames and Categories for the supplier 'Speciality  ↗
    Biscuits, Ltd.
508 USE AdventureWork;
509
510 SELECT
511     p.Name AS ProductName,
512     pc.Name AS CategoryName
513 FROM
514     Production.Product p
515 JOIN
516     Production.ProductSubcategory psc ON p.ProductSubcategoryID =  ↗
        psc.ProductSubcategoryID
517 JOIN
518     Production.ProductCategory pc ON psc.ProductCategoryID =  ↗
        pc.ProductCategoryID
519 JOIN
520     Purchasing.ProductVendor pv ON p.ProductID = pv.ProductID
521 JOIN
522     Purchasing.Vendor v ON pv.BusinessEntityID = v.BusinessEntityID
523 WHERE
524     v.Name = 'Speciality Biscuits, Ltd.';
525
526
527
528 --34-List of products that were never ordered.
529 USE AdventureWork;
530
531 SELECT
532     p.ProductID,
533     p.Name AS ProductName
534 FROM
535     Production.Product p
536 LEFT JOIN
537     Sales.SalesOrderDetail sod ON p.ProductID = sod.ProductID
538 WHERE
539     sod.ProductID IS NULL;
540
541
542
543 --35--List of products where units in stock is less than 10 and units  ↗
    on order are 0.
544 USE AdventureWork;
545
546 SELECT
547     p.ProductID,
548     p.Name AS ProductName,
549     pi.Quantity AS UnitsInStock
550 FROM
551     Production.Product p
552 JOIN
553     Production.ProductInventory pi ON p.ProductID = pi.ProductID
554 LEFT JOIN
```

```
555 (SELECT ProductID, SUM(OrderQty) AS TotalOrdered
556 FROM Purchasing.PurchaseOrderDetail
557 GROUP BY ProductID) pod ON p.ProductID = pod.ProductID
558 WHERE
559 pi.Quantity < 10
560 AND (pod.TotalOrdered IS NULL OR pod.TotalOrdered = 0);
561
562
563
564 --36-List of top 10 countries by sales.
565 USE AdventureWork;
566
567 SELECT TOP 10
568 cr.Name AS Country,
569 SUM(soh.TotalDue) AS TotalSales
570 FROM
571 Sales.SalesOrderHeader soh
572 JOIN
573 Person.Address a ON soh.ShipToAddressID = a.AddressID
574 JOIN
575 Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
576 JOIN
577 Person.CountryRegion cr ON sp.CountryRegionCode =
    cr.CountryRegionCode
578 GROUP BY
579 cr.Name
580 ORDER BY
581 TotalSales DESC;
582
583
584
585 --37-Number of orders each employee has taken for customers with
    CustomerIDs between A and AO.
586 USE AdventureWork;
587
588 SELECT
589 e.BusinessEntityID AS EmployeeID,
590 COUNT(soh.SalesOrderID) AS OrderCount
591 FROM
592 Sales.SalesOrderHeader soh
593 JOIN
594 Sales.Customer c ON soh.CustomerID = c.CustomerID
595 JOIN
596 HumanResources.Employee e ON soh.SalesPersonID = e.BusinessEntityID
597 WHERE
598 c.AccountNumber BETWEEN 'A' AND 'AO'
599 GROUP BY
600 e.BusinessEntityID;
601
602
603
604 --38-Orderdate of most expensive order.
605 USE AdventureWork;
```

```
606
607 SELECT
608     OrderDate
609 FROM
610     Sales.SalesOrderHeader
611 WHERE
612     TotalDue = (SELECT MAX(TotalDue) FROM Sales.SalesOrderHeader);
613
614
615
616 --39-Product name and total revenue from that product.
617 USE AdventureWork;
618
619 SELECT
620     p.Name AS ProductName,
621     SUM(sod.LineTotal) AS TotalRevenue
622 FROM
623     Sales.SalesOrderDetail sod
624 JOIN
625     Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
626 JOIN
627     Production.Product p ON sod.ProductID = p.ProductID
628 GROUP BY
629     p.Name
630 ORDER BY
631     TotalRevenue DESC;
632
633
634
635 --40-Supplierid and number of products offered.
636 USE AdventureWork;
637
638 SELECT
639     pv.BusinessEntityID AS SupplierID,
640     COUNT(pv.ProductID) AS NumberOfProducts
641 FROM
642     Purchasing.ProductVendor pv
643 GROUP BY
644     pv.BusinessEntityID
645 ORDER BY
646     NumberOfProducts DESC;
647
648
649
650 --41-Top ten customers based on their business.
651 USE AdventureWork;
652
653 SELECT
654     c.CustomerID,
655     p.FirstName,
656     p.LastName,
657     SUM(soh.TotalDue) AS TotalBusiness
658 FROM
```

```
659     Sales.SalesOrderHeader soh
660 JOIN
661     Sales.Customer c ON soh.CustomerID = c.CustomerID
662 JOIN
663     Person.Person p ON c.PersonID = p.BusinessEntityID
664 GROUP BY
665     c.CustomerID,
666     p.FirstName,
667     p.LastName
668 ORDER BY
669     TotalBusiness DESC
670 OFFSET 0 ROWS
671 FETCH NEXT 10 ROWS ONLY;
672
673
674
675 --42-what is the total revenue of that company.
676 USE AdventureWork;
677
678 SELECT
679     SUM(TotalDue) AS TotalRevenue
680 FROM
681     Sales.SalesOrderHeader;
```