

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

```
44 WHERE Name LIKE 'A%'
45 ORDER BY Name;
46
47
48
49 --7-List of customers who ever placed an order
50 SELECT DISTINCT c.*
51 FROM Sales.Customer c
52 JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID;
53
54
55
56 --8-List of customers who live in london and have bought chai
57 SELECT DISTINCT c.*
58 FROM Sales.Customer c
59 JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID
60 JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
61 JOIN Production.Product p ON sod.ProductID = p.ProductID
62 JOIN Person.BusinessEntityAddress bea ON soh.ShipToAddressID =
    bea.BusinessEntityID
63 JOIN Person.Address a ON bea.AddressID = a.AddressID
64 WHERE a.City = 'London' AND p.Name = 'Chai';
65
66
67
68 --9-List of customer who never placed an order
69 SELECT * FROM Sales.Customer c
70 WHERE NOT EXISTS (
71     SELECT 1
72     FROM Sales.SalesOrderHeader soh
73     WHERE c.CustomerID = soh.CustomerID
74 );
75
76
77
78 --10-List of customers who ordered a Tofu
79 SELECT DISTINCT c.*
80 FROM Sales.Customer c
81 JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID
82 JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
83 JOIN Production.Product p ON sod.ProductID = p.ProductID
84 WHERE p.Name = 'Tofu';
85
86
87
88 --11-Details of first order of the system
89 SELECT TOP 1 *
90 FROM Sales.SalesOrderHeader
91 ORDER BY OrderDate ASC;
```

```
92
93
94
95 --12-Find the details of most expensive order date
96 WITH OrderTotals AS (
97     SELECT
98         SalesOrderID,
99         SUM(LineTotal) AS TotalOrderCost
100     FROM
101         Sales.SalesOrderDetail
102     GROUP BY
103         SalesOrderID
104 )
105 SELECT
106     soh.SalesOrderID,
107     soh.OrderDate,
108     soh.DueDate,
109     soh.ShipDate,
110     soh.Status,
111     soh.OnlineOrderFlag,
112     soh.SalesOrderNumber,
113     soh.PurchaseOrderNumber,
114     soh.AccountNumber,
115     soh.CustomerID,
116     soh.SalesPersonID,
117     soh.TerritoryID,
118     soh.BillToAddressID,
119     soh.ShipToAddressID,
120     soh.ShipMethodID,
121     soh.CreditCardID,
122     soh.CreditCardApprovalCode,
123     soh.CurrencyRateID,
124     soh.SubTotal,
125     soh.TaxAmt,
126     soh.Freight,
127     soh.TotalDue,
128     soh.Comment,
129     soh.rowguid,
130     soh.ModifiedDate,
131     ot.TotalOrderCost
132 FROM
133     Sales.SalesOrderHeader soh
134 JOIN
135     OrderTotals ot ON soh.SalesOrderID = ot.SalesOrderID
136 ORDER BY
137     ot.TotalOrderCost DESC
138
139
140
```

```
141 --13- for each order get the orderid and average quantity of items in that order
142 SELECT
143     SalesOrderID,
144     AVG(OrderQty) AS AverageQuantity
145 FROM
146     Sales.SalesOrderDetail
147 GROUP BY
148     SalesOrderID;
149
150
151
152 --14-For each order get the OrderID,minimum quantity and maximum quantity for that order
153 SELECT
154     SalesOrderID,
155     MIN(OrderQty) AS MinimumQuantity,
156     MAX(OrderQty) AS MaximumQuantity
157 FROM
158     Sales.SalesOrderDetail
159 GROUP BY
160     SalesOrderID;
161
162
163
164 --15-Get a list of all managers and total no of employees who report to them.
165 USE AdventureWork;
166
167 SELECT
168     E.BusinessEntityID AS ManagerID,
169     P.FirstName + ' ' + P.LastName AS ManagerName,
170     COUNT(RE.BusinessEntityID) AS TotalEmployees
171 FROM
172     HumanResources.Employee AS E
173 INNER JOIN
174     Person.Person AS P ON E.BusinessEntityID = P.BusinessEntityID
175 LEFT JOIN
176     HumanResources.Employee AS RE ON RE.OrganizationNode.GetAncestor(1) = E.OrganizationNode
177 GROUP BY
178     E.BusinessEntityID, P.FirstName, P.LastName
179 ORDER BY
180     TotalEmployees DESC;
181
182
183
184 --16-Get the orderID and total quantity for each order that has a total quantity of greater than 300.
```

```
185 SELECT
186     SalesOrderID,
187     SUM(OrderQty) AS TotalQuantity
188 FROM
189     Sales.SalesOrderDetail
190 GROUP BY
191     SalesOrderID
192 HAVING
193     SUM(OrderQty) > 300;
194
195 --17-List of all the orders placed on after 1996/12/31.
196 SELECT *
197 FROM Sales.SalesOrderHeader
198 WHERE OrderDate >= '1996-12-31';
199
200 --18-List of all orders shipped to Canada.
201 SELECT
202     soh.SalesOrderID,
203     soh.OrderDate,
204     soh.ShipDate,
205     soh.Status,
206     soh.TotalDue,
207     a.AddressLine1,
208     a.AddressLine2,
209     a.City,
210     sp.Name AS StateProvince,
211     cr.Name AS CountryRegion
212 FROM
213     Sales.SalesOrderHeader soh
214 JOIN
215     Person.Address a ON soh.ShipToAddressID = a.AddressID
216 JOIN
217     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
218 JOIN
219     Person.CountryRegion cr ON sp.CountryRegionCode = cr.CountryRegionCode
220 WHERE
221     cr.Name = 'Canada';
222
223
224
225 --19-List of all orders with order total > 200 .
226 SELECT *
227 FROM Sales.SalesOrderHeader
228 WHERE TotalDue > 200;
229
230 --20-List of all countries and sales made in each country .
231 SELECT
232     cr.Name AS Country,
233     SUM(soh.TotalDue) AS TotalSales
```

```
234 FROM
235     Sales.SalesOrderHeader soh
236 JOIN
237     Person.Address a ON soh.ShipToAddressID = a.AddressID
238 JOIN
239     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
240 JOIN
241     Person.CountryRegion cr ON sp.CountryRegionCode = cr.CountryRegionCode
242 GROUP BY
243     cr.Name
244 ORDER BY
245     TotalSales DESC;
246
247
248
249 --21-List of customers Contact name and numbers of orders they placed .
250 SELECT
251     p.FirstName + ' ' + p.LastName AS ContactName,
252     COUNT(soh.SalesOrderID) AS NumberOfOrders
253 FROM
254     Sales.SalesOrderHeader soh
255 JOIN
256     Sales.Customer c ON soh.CustomerID = c.CustomerID
257 JOIN
258     Person.Person p ON c.PersonID = p.BusinessEntityID
259 GROUP BY
260     p.FirstName, p.LastName
261 ORDER BY
262     NumberOfOrders DESC;
263
264
265
266 --22-List of customers contactnames who have placed more than 3 orders.
267 USE AdventureWork;
268
269 SELECT
270     P.FirstName + ' ' + P.LastName AS ContactName,
271     COUNT(SOH.SalesOrderID) AS NumberOfOrders
272 FROM
273     Sales.Customer AS C
274 INNER JOIN
275     Sales.SalesOrderHeader AS SOH ON C.CustomerID = SOH.CustomerID
276 INNER JOIN
277     Person.Person AS P ON C.PersonID = P.BusinessEntityID
278 GROUP BY
279     P.FirstName, P.LastName
280 HAVING
281     COUNT(SOH.SalesOrderID) > 3
282 ORDER BY
```

```
283     NumberOfOrders DESC;
284
285
286
287 --23-List of discontinued products which were ordered between 1/1/1997      ↗
    and 1/1/1998
288 USE AdventureWork;
289
290 SELECT
291     P.ProductID,
292     P.Name AS ProductName,
293     P.DiscontinuedDate
294 FROM
295     Sales.SalesOrderDetail AS SOD
296 INNER JOIN
297     Sales.SalesOrderHeader AS SOH ON SOD.SalesOrderID = 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
303     AND P.DiscontinuedDate <= '1998-01-01'
304 GROUP BY
305     P.ProductID, P.Name, P.DiscontinuedDate
306 ORDER BY
307     P.ProductID;
308
309
310
311 --24-List of employees firstname,lastname,supervisor Firstname,Lastname
312 USE AdventureWork;
313
314 SELECT
315     E1.FirstName AS EmployeeFirstName,
316     E1.LastName AS EmployeeLastName,
317     E2.FirstName AS SupervisorFirstName,
318     E2.LastName AS SupervisorLastName
319 FROM
320     HumanResources.Employee AS Emp1
321 INNER JOIN
322     Person.Person AS E1 ON Emp1.BusinessEntityID = E1.BusinessEntityID
323 LEFT JOIN
324     HumanResources.Employee AS Emp2 ON Emp1.OrganizationNode.GetAncestor      ↗
        (1) = Emp2.OrganizationNode
325 LEFT JOIN
326     Person.Person AS E2 ON Emp2.BusinessEntityID = E2.BusinessEntityID
327 ORDER BY
328     EmployeeLastName, EmployeeFirstName;
329
```

```
330
331
332 --25-List of employees ID and total sale conducted by employee.
333 USE AdventureWork;
334
335 SELECT
336     E.BusinessEntityID AS EmployeeID,
337     SUM(SOH.TotalDue) AS TotalSales
338 FROM
339     Sales.SalesOrderHeader AS SOH
340 INNER JOIN
341     Sales.SalesPerson AS SP ON SOH.SalesPersonID = SP.BusinessEntityID
342 INNER JOIN
343     HumanResources.Employee AS E ON SP.BusinessEntityID =
        E.BusinessEntityID
344 GROUP BY
345     E.BusinessEntityID
346 ORDER BY
347     TotalSales DESC;
348
349
350
351 --26-List of employees whose Firstname contains character a.
352 USE AdventureWork;
353 SELECT FirstName, LastName
354 FROM HumanResources.Employee AS e
355 JOIN Person.Person AS p ON e.BusinessEntityID = p.BusinessEntityID
356 WHERE p.FirstName LIKE '%a%';
357
358
359
360 --27-List of managers who have more than four people reporting to them.
361 USE AdventureWork;
362
363 SELECT
364     ManagerPerson.FirstName,
365     ManagerPerson.LastName,
366     COUNT(Employee.BusinessEntityID) AS ReportCount
367 FROM
368     HumanResources.Employee AS Employee
369 JOIN
370     HumanResources.Employee AS Manager ON
        Employee.OrganizationNode.GetAncestor(1) = Manager.OrganizationNode
371 JOIN
372     Person.Person AS ManagerPerson ON Manager.BusinessEntityID =
        ManagerPerson.BusinessEntityID
373 GROUP BY
374     ManagerPerson.FirstName,
375     ManagerPerson.LastName
```



```
376 HAVING
377     COUNT(Employee.BusinessEntityID) > 4;
378
379
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
396 --29-List of orders placed by best customer.
397
398     --identify the best customer.
399     USE AdventureWork;
400 WITH CustomerTotal AS (
401     SELECT
402         soh.CustomerID,
403         SUM(soh.TotalDue) AS TotalSpent
404     FROM
405         Sales.SalesOrderHeader soh
406     GROUP BY
407         soh.CustomerID
408 )
409 SELECT TOP 1
410     CustomerID,
411     TotalSpent
412 FROM
413     CustomerTotal
414 ORDER BY
415     TotalSpent DESC;
416
417 --order placed by the best customer.
418 USE AdventureWork;
419 SELECT
420     soh.SalesOrderID,
421     soh.OrderDate,
422     soh.TotalDue
423 FROM
424     Sales.SalesOrderHeader soh
```

```
425 WHERE
426     soh.CustomerID = (
427         SELECT TOP 1
428             CustomerID
429         FROM
430             (
431                 SELECT
432                     soh.CustomerID,
433                     SUM(soh.TotalDue) AS TotalSpent
434                 FROM
435                     Sales.SalesOrderHeader soh
436                 GROUP BY
437                     soh.CustomerID
438             ) AS CustomerTotal
439         ORDER BY
440             TotalSpent DESC
441     );
442
443 --30-List of orders placed by customer who do not have a fax number.
444 USE AdventureWork;
445
446 SELECT
447     soh.SalesOrderID,
448     soh.OrderDate,
449     soh.TotalDue,
450     c.CustomerID,
451     p.FirstName,
452     p.LastName
453 FROM
454     Sales.SalesOrderHeader soh
455 JOIN
456     Sales.Customer c ON soh.CustomerID = c.CustomerID
457 JOIN
458     Person.Person p ON c.PersonID = p.BusinessEntityID
459 LEFT JOIN
460     Person.PersonPhone pp ON p.BusinessEntityID = pp.BusinessEntityID AND
461         pp.PhoneNumberTypeID = (SELECT PhoneNumberTypeID FROM
462             Person.PhoneNumberType WHERE Name = 'Fax')
463
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
```

```

472 JOIN
473     Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
474 JOIN
475     Production.Product p ON sod.ProductID = p.ProductID
476 JOIN
477     Person.Address a ON soh.ShipToAddressID = a.AddressID
478 WHERE
479     p.Name = 'Tofu';
480
481
482
483 --32-List of product Names that were shipped to France.
484 USE AdventureWork;
485
486 SELECT DISTINCT
487     p.Name AS ProductName
488 FROM
489     Sales.SalesOrderDetail sod
490 JOIN
491     Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
492 JOIN
493     Production.Product p ON sod.ProductID = p.ProductID
494 JOIN
495     Person.Address a ON soh.ShipToAddressID = a.AddressID
496 JOIN
497     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
498 JOIN
499     Person.CountryRegion cr ON sp.CountryRegionCode = cr.CountryRegionCode
500 WHERE
501     cr.Name = 'France';
502
503
504
505 --33-List of ProductNames and Categories for the supplier 'Speciality Biscuits, Ltd.
506 USE AdventureWork;
507
508 SELECT
509     p.Name AS ProductName,
510     pc.Name AS CategoryName
511 FROM
512     Production.Product p
513 JOIN
514     Production.ProductSubcategory psc ON p.ProductSubcategoryID =
        psc.ProductSubcategoryID
515 JOIN
516     Production.ProductCategory pc ON psc.ProductCategoryID =
        pc.ProductCategoryID
517 JOIN

```

```
518     Purchasing.ProductVendor pv ON p.ProductID = pv.ProductID
519 JOIN
520     Purchasing.Vendor v ON pv.BusinessEntityID = v.BusinessEntityID
521 WHERE
522     v.Name = 'Speciality Biscuits, Ltd.';
523
524
525
526 --34--List of products that were never ordered.
527 USE AdventureWork;
528
529 SELECT
530     p.ProductID,
531     p.Name AS ProductName
532 FROM
533     Production.Product p
534 LEFT JOIN
535     Sales.SalesOrderDetail sod ON p.ProductID = sod.ProductID
536 WHERE
537     sod.ProductID IS NULL;
538
539
540
541 --35--List of products where units in stock is less than 10 and units on  ↗
542     order are 0.
543
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 --36--List of top 10 countries by sales.
563 USE AdventureWork;
564
565 SELECT TOP 10
```

```
566     cr.Name AS Country,
567     SUM(soh.TotalDue) AS TotalSales
568 FROM
569     Sales.SalesOrderHeader soh
570 JOIN
571     Person.Address a ON soh.ShipToAddressID = a.AddressID
572 JOIN
573     Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
574 JOIN
575     Person.CountryRegion cr ON sp.CountryRegionCode = cr.CountryRegionCode
576 GROUP BY
577     cr.Name
578 ORDER BY
579     TotalSales DESC;
580
581 --37-Number of orders each employee has taken for customers with      ↗
582     CustomerIDs between A and AO.
583
584 USE AdventureWork;
585
586 SELECT
587     e.BusinessEntityID AS EmployeeID,
588     COUNT(soh.SalesOrderID) AS OrderCount
589 FROM
590     Sales.SalesOrderHeader soh
591 JOIN
592     Sales.Customer c ON soh.CustomerID = c.CustomerID
593 JOIN
594     HumanResources.Employee e ON soh.SalesPersonID = e.BusinessEntityID
595 WHERE
596     c.AccountNumber BETWEEN 'A' AND 'AO'
597 GROUP BY
598     e.BusinessEntityID;
599
600 --38-Orderdate of most expensive order.
601
602 USE AdventureWork;
603
604 SELECT
605     OrderDate
606 FROM
607     Sales.SalesOrderHeader
608 WHERE
609     TotalDue = (SELECT MAX(TotalDue) FROM Sales.SalesOrderHeader);
610
611 --39-Product name and total revenue from that product.
612
613 USE AdventureWork;
```

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

```
663 ORDER BY
664     TotalBusiness DESC
665 OFFSET 0 ROWS
666 FETCH NEXT 10 ROWS ONLY;
667
668 --42-what is the total revenue of that company.
669 USE AdventureWork;
670
671 SELECT
672     SUM(TotalDue) AS TotalRevenue
673 FROM
674     Sales.SalesOrderHeader;
```