

```

1  /* *****
2      INTRODUCTION TO STRUCTURED QUERY LANGUAGE FOR DATA ANALYTICS
3          WS24SQL10001, 2024/03/11 - 2024/04/10
4          https://folvera.commons.gc.cuny.edu/?cat=34
5  *****
6
7  SESSION #5 (2024/03/25): MANIPULATING DATA
8
9  1. Using clauses `BETWEEN`, `NOT`, `UNION`, `EXCEPT` and `INTERSECT`
10 2. Understanding function `FORMAT()` for dates and currencies including
11    culture codes
12 *****
13
14 1. LAB #3
15    Write a query
16    1.01. to call all columns and values from `AP1.Vendors` and any related
17          values from `AP1.ContactUpdates` (`LEFT JOIN`),
18    1.02. to put together `FirstName` and `LastName` in one field with alias
19          `ContactName`,
20    1.03. to put together the first letter of `FirstName`, the complete
21          `LastName`, `@`, `VendorName` (removing empty spaces between words and
22          special characters like `&` and `,`) and `.com` as `ContactEmail`
23          presenting the output in lower case,
24    1.04. and to put together `VendorContactFName` and `VendorContactLName`
25          with aliases `VendorContactName` and `VendorContactEmail` (like
26          #1.02).
27 ***** */
28
29 SELECT AP1.Vendors.VendorID,
30        AP1.Vendors.VendorName,
31        AP1.Vendors.VendorAddress1,
32        AP1.Vendors.VendorAddress2,
33        AP1.Vendors.VendorCity,
34        AP1.Vendors.VendorState,
35        AP1.Vendors.VendorZipCode,
36        AP1.Vendors.VendorPhone,
37        AP1.Vendors.VendorContactFName +      -- 1. concatenation of
38        ' ' +                                  --   `FirstName`, a single
39        AP1.Vendors.VendorContactLName        --   space and `LastName`
40        AS ContactName,                      --   using `+` with alias
41                                                --   `ContactName`
42        LEFT(AP1.Vendors.VendorContactFName, 1) + -- 2. concatenation of
43        AP1.Vendors.VendorContactLName +      --   `VendorContactFName`,
44        '@' +                                  --   a single space and
45                                                --   `VendorContactLName`,
46                                                --   the `@` sign, followed by
47                                                --   `VendorName` (value #6)
48        REPLACE(                              -- 2.01. getting value #5
49            REPLACE(                            -- 2.02. getting value #4
50                REPLACE(                      -- 2.03. getting value #3
51                    REPLACE(                -- 2.04. getting value #2
52                        REPLACE(            -- 2.05. getting value #1
53                            REPLACE(AP1.Vendors.VendorName, ' ', ''))
54                                -- 2.06. generating value #1
55                                -- 2.07. generating value #2
56                                -- 2.08. generating value #3
57                                -- 2.09. generating value #4
58                                -- 2.10. generating value #5
59                                -- 2.11. generating value #6
60                                --   where `''`
61                                --   represents a single
62                                --   quote (`'`) used an
63                                --   escape character
64                                --   and `.foo` to complete
65                                --   email with alias
66                                --
67                                --
68                                --
69                                --
70                                --
71                                --
72                                --
73                                --
74                                --
75                                --
76                                --
77                                --
78                                --
79                                --
80                                --
81                                --
82                                --
83                                --
84                                --
85                                --
86                                --
87                                --
88                                --
89                                --
90                                --
91                                --
92                                --
93                                --
94                                --
95                                --
96                                --
97                                --
98                                --
99                                --
100                               --
101                               --
102                               --
103                               --
104                               --
105                               --
106                               --
107                               --
108                               --
109                               --
110                               --
111                               --
112                               --
113                               --
114                               --
115                               --
116                               --
117                               --
118                               --
119                               --
120                               --
121                               --
122                               --
123                               --
124                               --
125                               --
126                               --
127                               --
128                               --
129                               --
130                               --
131                               --
132                               --
133                               --
134                               --
135                               --
136                               --
137                               --
138                               --
139                               --
140                               --
141                               --
142                               --
143                               --
144                               --
145                               --
146                               --
147                               --
148                               --
149                               --
150                               --
151                               --
152                               --
153                               --
154                               --
155                               --
156                               --
157                               --
158                               --
159                               --
160                               --
161                               --
162                               --
163                               --
164                               --
165                               --
166                               --
167                               --
168                               --
169                               --
170                               --
171                               --
172                               --
173                               --
174                               --
175                               --
176                               --
177                               --
178                               --
179                               --
180                               --
181                               --
182                               --
183                               --
184                               --
185                               --
186                               --
187                               --
188                               --
189                               --
190                               --
191                               --
192                               --
193                               --
194                               --
195                               --
196                               --
197                               --
198                               --
199                               --
200                               --
201                               --
202                               --
203                               --
204                               --
205                               --
206                               --
207                               --
208                               --
209                               --
210                               --
211                               --
212                               --
213                               --
214                               --
215                               --
216                               --
217                               --
218                               --
219                               --
220                               --
221                               --
222                               --
223                               --
224                               --
225                               --
226                               --
227                               --
228                               --
229                               --
230                               --
231                               --
232                               --
233                               --
234                               --
235                               --
236                               --
237                               --
238                               --
239                               --
240                               --
241                               --
242                               --
243                               --
244                               --
245                               --
246                               --
247                               --
248                               --
249                               --
250                               --
251                               --
252                               --
253                               --
254                               --
255                               --
256                               --
257                               --
258                               --
259                               --
260                               --
261                               --
262                               --
263                               --
264                               --
265                               --
266                               --
267                               --
268                               --
269                               --
270                               --
271                               --
272                               --
273                               --
274                               --
275                               --
276                               --
277                               --
278                               --
279                               --
280                               --
281                               --
282                               --
283                               --
284                               --
285                               --
286                               --
287                               --
288                               --
289                               --
290                               --
291                               --
292                               --
293                               --
294                               --
295                               --
296                               --
297                               --
298                               --
299                               --
300                               --
301                               --
302                               --
303                               --
304                               --
305                               --
306                               --
307                               --
308                               --
309                               --
310                               --
311                               --
312                               --
313                               --
314                               --
315                               --
316                               --
317                               --
318                               --
319                               --
320                               --
321                               --
322                               --
323                               --
324                               --
325                               --
326                               --
327                               --
328                               --
329                               --
330                               --
331                               --
332                               --
333                               --
334                               --
335                               --
336                               --
337                               --
338                               --
339                               --
340                               --
341                               --
342                               --
343                               --
344                               --
345                               --
346                               --
347                               --
348                               --
349                               --
350                               --
351                               --
352                               --
353                               --
354                               --
355                               --
356                               --
357                               --
358                               --
359                               --
360                               --
361                               --
362                               --
363                               --
364                               --
365                               --
366                               --
367                               --
368                               --
369                               --
370                               --
371                               --
372                               --
373                               --
374                               --
375                               --
376                               --
377                               --
378                               --
379                               --
380                               --
381                               --
382                               --
383                               --
384                               --
385                               --
386                               --
387                               --
388                               --
389                               --
390                               --
391                               --
392                               --
393                               --
394                               --
395                               --
396                               --
397                               --
398                               --
399                               --
400                               --
401                               --
402                               --
403                               --
404                               --
405                               --
406                               --
407                               --
408                               --
409                               --
410                               --
411                               --
412                               --
413                               --
414                               --
415                               --
416                               --
417                               --
418                               --
419                               --
420                               --
421                               --
422                               --
423                               --
424                               --
425                               --
426                               --
427                               --
428                               --
429                               --
430                               --
431                               --
432                               --
433                               --
434                               --
435                               --
436                               --
437                               --
438                               --
439                               --
440                               --
441                               --
442                               --
443                               --
444                               --
445                               --
446                               --
447                               --
448                               --
449                               --
450                               --
451                               --
452                               --
453                               --
454                               --
455                               --
456                               --
457                               --
458                               --
459                               --
460                               --
461                               --
462                               --
463                               --
464                               --
465                               --
466                               --
467                               --
468                               --
469                               --
470                               --
471                               --
472                               --
473                               --
474                               --
475                               --
476                               --
477                               --
478                               --
479                               --
480                               --
481                               --
482                               --
483                               --
484                               --
485                               --
486                               --
487                               --
488                               --
489                               --
490                               --
491                               --
492                               --
493                               --
494                               --
495                               --
496                               --
497                               --
498                               --
499                               --
500                               --
501                               --
502                               --
503                               --
504                               --
505                               --
506                               --
507                               --
508                               --
509                               --
510                               --
511                               --
512                               --
513                               --
514                               --
515                               --
516                               --
517                               --
518                               --
519                               --
520                               --
521                               --
522                               --
523                               --
524                               --
525                               --
526                               --
527                               --
528                               --
529                               --
530                               --
531                               --
532                               --
533                               --
534                               --
535                               --
536                               --
537                               --
538                               --
539                               --
540                               --
541                               --
542                               --
543                               --
544                               --
545                               --
546                               --
547                               --
548                               --
549                               --
550                               --
551                               --
552                               --
553                               --
554                               --
555                               --
556                               --
557                               --
558                               --
559                               --
560                               --
561                               --
562                               --
563                               --
564                               --
565                               --
566                               --
567                               --
568                               --
569                               --
570                               --
571                               --
572                               --
573                               --
574                               --
575                               --
576                               --
577                               --
578                               --
579                               --
580                               --
581                               --
582                               --
583                               --
584                               --
585                               --
586                               --
587                               --
588                               --
589                               --
590                               --
591                               --
592                               --
593                               --
594                               --
595                               --
596                               --
597                               --
598                               --
599                               --
600                               --
601                               --
602                               --
603                               --
604                               --
605                               --
606                               --
607                               --
608                               --
609                               --
610                               --
611                               --
612                               --
613                               --
614                               --
615                               --
616                               --
617                               --
618                               --
619                               --
620                               --
621                               --
622                               --
623                               --
624                               --
625                               --
626                               --
627                               --
628                               --
629                               --
630                               --
631                               --
632                               --
633                               --
634                               --
635                               --
636                               --
637                               --
638                               --
639                               --
640                               --
641                               --
642                               --
643                               --
644                               --
645                               --
646                               --
647                               --
648                               --
649                               --
650                               --
651                               --
652                               --
653                               --
654                               --
655                               --
656                               --
657                               --
658                               --
659                               --
660                               --
661                               --
662                               --
663                               --
664                               --
665                               --
666                               --
667                               --
668                               --
669                               --
670                               --
671                               --
672                               --
673                               --
674                               --
675                               --
676                               --
677                               --
678                               --
679                               --
680                               --
681                               --
682                               --
683                               --
684                               --
685                               --
686                               --
687                               --
688                               --
689                               --
690                               --
691                               --
692                               --
693                               --
694                               --
695                               --
696                               --
697                               --
698                               --
699                               --
700                               --
701                               --
702                               --
703                               --
704                               --
705                               --
706                               --
707                               --
708                               --
709                               --
710                               --
711                               --
712                               --
713                               --
714                               --
715                               --
716                               --
717                               --
718                               --
719                               --
720                               --
721                               --
722                               --
723                               --
724                               --
725                               --
726                               --
727                               --
728                               --
729                               --
730                               --
731                               --
732                               --
733                               --
734                               --
735                               --
736                               --
737                               --
738                               --
739                               --
740                               --
741                               --
742                               --
743                               --
744                               --
745                               --
746                               --
747                               --
748                               --
749                               --
750                               --
751                               --
752                               --
753                               --
754                               --
755                               --
756                               --
757                               --
758                               --
759                               --
760                               --
761                               --
762                               --
763                               --
764                               --
765                               --
766                               --
767                               --
768                               --
769                               --
770                               --
771                               --
772                               --
773                               --
774                               --
775                               --
776                               --
777                               --
778                               --
779                               --
780                               --
781                               --
782                               --
783                               --
784                               --
785                               --
786                               --
787                               --
788                               --
789                               --
790                               --
791                               --
792                               --
793                               --
794                               --
795                               --
796                               --
797                               --
798                               --
799                               --
800                               --
801                               --
802                               --
803                               --
804                               --
805                               --
806                               --
807                               --
808                               --
809                               --
810                               --
811                               --
812                               --
813                               --
814                               --
815                               --
816                               --
817                               --
818                               --
819                               --
820                               --
821                               --
822                               --
823                               --
824                               --
825                               --
826                               --
827                               --
828                               --
829                               --
830                               --
831                               --
832                               --
833                               --
834                               --
835                               --
836                               --
837                               --
838                               --
839                               --
840                               --
841                               --
842                               --
843                               --
844                               --
845                               --
846                               --
847                               --
848                               --
849                               --
850                               --
851                               --
852                               --
853                               --
854                               --
855                               --
856                               --
857                               --
858                               --
859                               --
860                               --
861                               --
862                               --
863                               --
864                               --
865                               --
866                               --
867                               --
868                               --
869                               --
870                               --
871                               --
872                               --
873                               --
874                               --
875                               --
876                               --
877                               --
878                               --
879                               --
880                               --
881                               --
882                               --
883                               --
884                               --
885                               --
886                               --
887                               --
888                               --
889                               --
890                               --
891                               --
892                               --
893                               --
894                               --
895                               --
896                               --
897                               --
898                               --
899                               --
900                               --
901                               --
902                               --
903                               --
904                               --
905                               --
906                               --
907                               --
908                               --
909                               --
910                               --
911                               --
912                               --
913                               --
914                               --
915                               --
916                               --
917                               --
918                               --
919                               --
920                               --
921                               --
922                               --
923                               --
924                               --
925                               --
926                               --
927                               --
928                               --
929                               --
930                               --
931                               --
932                               --
933                               --
934                               --
935                               --
936                               --
937                               --
938                               --
939                               --
940                               --
941                               --
942                               --
943                               --
944                               --
945                               --
946                               --
947                               --
948                               --
949                               --
950                               --
951                               --
952                               --
953                               --
954                               --
955                               --
956                               --
957                               --
958                               --
959                               --
960                               --
961                               --
962                               --
963                               --
964                               --
965                               --
966                               --
967                               --
968                               --
969                               --
970                               --
971                               --
972                               --
973                               --
974                               --
975                               --
976                               --
977                               --
978                               --
979                               --
980                               --
981                               --
982                               --
983                               --
984                               --
985                               --
986                               --
987                               --
988                               --
989                               --
990                               --
991                               --
992                               --
993                               --
994                               --
995                               --
996                               --
997                               --
998                               --
999                               --
1000                              --

```

```

66                                     --      `VendorContactEmail`
67     AP1.Vendors.DefaultTermsID,
68     AP1.Vendors.DefaultAccountNo,
69     -- AP1.ContactUpdates.VendorID AS Expr1,
70     AP1.ContactUpdates.FirstName +
71     ' ' +
72     AP1.ContactUpdates.LastName
73     AS ContactName,
74
75     LEFT(AP1.ContactUpdates.FirstName, 1) +
76     AP1.ContactUpdates.LastName +
77     '@' +
78
79                                     -- 3. concatenation of
80                                     --      `VendorContactFName`, a
81                                     --      single space and
82                                     --      `LastName` using `+` with
83                                     --      alias `VendorContactLName`
84
85                                     -- 4. concatenation of
86                                     --      `VendorContactFName`,
87                                     --      a single space and
88                                     --      `VendorContactLName`,
89                                     --      the `@` sign, followed by
90                                     --      `VendorName` (value #6)
91                                     --      4.01. getting value #5
92                                     --      4.02. getting value #4
93                                     --      4.03. getting value #3
94                                     --      4.04. getting value #2
95                                     --      4.05. getting value #1
96
97                                     --      4.06. generating value #1
98                                     --      4.07. generating value #2
99                                     --      4.08. generating value #3
100                                    --      4.09. generating value #4
101                                    --      4.10. generating value #5
102                                    --      4.11. generating value #6
103                                    --      where `''`
104                                    --      represents a single
105                                    --      quote (``) used an
106                                    --      escape character
107                                    --      and `.foo` to complete
108                                    --      email with alias
109                                    --      `VendorContactEmail`
110
111     REPLACE(
112     REPLACE(
113     REPLACE(
114     REPLACE(
115     REPLACE(
116     REPLACE(
117     REPLACE(
118     REPLACE(
119     REPLACE(
120     REPLACE(
121     REPLACE(
122     REPLACE(
123     REPLACE(
124     REPLACE(
125     REPLACE(
126     REPLACE(
127     REPLACE(
128     REPLACE(
129     REPLACE(
130     REPLACE(

```

```

/* *****
As with previous example, we can use an alias for each table, which
in this case, allows us to present neater code.

```

```

`v` for `AP1.Vendors`
`c` for `AP1.ContactUpdates`
***** */

```

```

113 SELECT v.VendorID,
114        v.VendorName,
115        v.VendorAddress1,
116        v.VendorAddress2,
117        v.VendorCity,
118        v.VendorState,
119        v.VendorZipCode,
120        v.VendorPhone,
121        v.VendorContactFName +
122        ' ' +
123        v.VendorContactLName AS ContactName,
124     LEFT(v.VendorContactFName, 1) +
125     v.VendorContactLName + '@' +
126     REPLACE(
127     REPLACE(
128     REPLACE(
129     REPLACE(
130     REPLACE(

```

```

131         REPLACE(v.VendorName, ' ', '')
132     , ' ', ''
133     , ' ', ''
134     , '/' , ''
135     , '&' , ''
136     , '''' , ''
137     + '.foo' AS VendorContactEmail,
138 v.DefaultTermsID,
139 v.DefaultAccountNo,
140 -- c.VendorID AS Expr1,
141 c.FirstName +
142 ' ' +
143 c.LastName AS ContactName,
144 LEFT(c.FirstName, 1) +
145 c.LastName +
146 '@' +
147 REPLACE(
148     REPLACE(
149         REPLACE(
150             REPLACE(
151                 REPLACE(
152                     REPLACE(v.VendorName, ' ', '')
153                 , ' ', ''
154             , ' ', ''
155         , '/' , ''
156     , '&' , ''
157     , '''' , ''
158     + '.foo' AS VendorContactEmail
159 FROM AP1.Vendors AS v
160 LEFT JOIN AP1.ContactUpdates AS c
161     ON v.VendorID = c.VendorID;
162
163
164 /* *****
165     Instead of using a plus sign (`+`), we can use `CONCAT()` since
166     adding a value and NULL returns NULL. In other words, we lose data,
167     which would be logical error (not syntax error).
168     ***** */
169
170 SELECT AP1.Vendors.VendorID,
171     AP1.Vendors.VendorName,
172     AP1.Vendors.VendorAddress1,
173     AP1.Vendors.VendorAddress2,
174     AP1.Vendors.VendorCity,
175     AP1.Vendors.VendorState,
176     AP1.Vendors.VendorZipCode,
177     AP1.Vendors.VendorPhone,
178     CONCAT (
179         AP1.Vendors.VendorContactFName,
180         ' ',
181         AP1.Vendors.VendorContactLName
182     ) AS ContactName,
183     CONCAT (
184         LEFT(AP1.Vendors.VendorContactFName, 1),
185         AP1.Vendors.VendorContactLName,
186         '@',
187         REPLACE(
188             REPLACE(
189                 REPLACE(
190                     REPLACE(
191                         REPLACE(
192                             REPLACE(
193                                 REPLACE(AP1.Vendors.VendorName, ' ', '')
194                             , ' ', ''
195                             , ' ', ''
196                             , '/' , ''
197                             , '&' , ''
198                             , '''' , ''
199                             + '.foo' AS VendorContactEmail
200     ) AS VendorContactEmail,
201     v.DefaultTermsID,
202     v.DefaultAccountNo,
203     -- c.VendorID AS Expr1,
204     c.FirstName +
205     ' ' +
206     c.LastName AS ContactName,
207     LEFT(c.FirstName, 1) +
208     c.LastName +
209     '@' +
210     REPLACE(
211         REPLACE(
212             REPLACE(
213                 REPLACE(
214                     REPLACE(
215                         REPLACE(v.VendorName, ' ', '')
216                     , ' ', ''
217                 , ' ', ''
218             , '/' , ''
219         , '&' , ''
220         , '''' , ''
221         + '.foo' AS VendorContactEmail
222 FROM AP1.Vendors AS v
223 LEFT JOIN AP1.ContactUpdates AS c
224     ON v.VendorID = c.VendorID;
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295

```

```

-- 1. concatenation of
-- `FirstName`, a single
-- space and `LastName` with
-- alias `ContactName`
-- 2. concatenation of
-- `VendorContactFName`,
-- a single space and
-- `VendorContactLName`,
-- the `@` sign, followed by
-- `VendorName` (value #6)
-- 2.01. getting value #5
-- 2.02. getting value #4
-- 2.03. getting value #3
-- 2.04. getting value #2
-- 2.05. getting value #1
-- 2.06. generating value #1

```

```

196         , ' ', ''')
197     , ' ', ''')
198     , ' / ', ''')
199     , ' & ', ''')
200     , ' ', ''')
201
202
203
204
205     , '.foo'
206 ) AS VendorContactEmail,
207
208 AP1.Vendors.DefaultTermsID,
209 AP1.Vendors.DefaultAccountNo,
210 -- AP1.ContactUpdates VendorID AS Expr1,
211 CONCAT (
212     AP1.ContactUpdates.FirstName,
213     ' ',
214     AP1.ContactUpdates.LastName
215 ) AS ContactName,
216 CONCAT (
217     LEFT(AP1.ContactUpdates.FirstName, 1),
218     AP1.ContactUpdates.LastName,
219     '@',
220
221     REPLACE(
222         REPLACE(
223             REPLACE(
224                 REPLACE(
225                     REPLACE(
226                         REPLACE(
227                             REPLACE(AP1.Vendors.VendorName, ' ', ''')
228                             , ' ', ''')
229                             , ' ', ''')
230                             , ' / ', ''')
231                             , ' & ', ''')
232                             , ' ', '''),
233     , ' ', '''),
234
235
236
237
238     '.foo'
239 ) AS VendorContactEmail
240
241 FROM AP1.Vendors
242 LEFT JOIN AP1.ContactUpdates
243 ON AP1.Vendors.VendorID = AP1.ContactUpdates.VendorID;
244
245
246 /* *****
247     Using `CONCAT()` also returns a logical error (not syntax error)
248     since the concatenation to make the second email returns values like
249     `@USPostalService.foo` since there is no corresponding `FirstName`
250     and `LastName`. We could use a CASE clause.
251 ***** */
252
253 SELECT AP1.Vendors.VendorID,
254     AP1.Vendors.VendorName,
255     AP1.Vendors.VendorAddress1,
256     AP1.Vendors.VendorAddress2,
257     AP1.Vendors.VendorCity,
258     AP1.Vendors.VendorState,
259     AP1.Vendors.VendorZipCode,
260     AP1.Vendors.VendorPhone,

```

```

-- 2.07. generating value #2
-- 2.08. generating value #3
-- 2.09. generating value #4
-- 2.10. generating value #5
-- 2.11. generating value #6
-- where ` `
--     represents a single
--     quote (` `) used an
--     escape character
-- and `.foo` to complete
-- email with alias
-- `VendorContactEmail`
--
-- 3. concatenation of
-- `VendorContactFName`, a
-- single space and
-- `LastName` with alias
-- `VendorContactLName`
--
-- 4. concatenation of
-- `VendorContactFName`,
-- a single space and
-- `VendorContactLName`,
-- the `@` sign, followed by
-- `VendorName` (value #6)
--
-- 4.01. getting value #5
-- 4.02. getting value #4
-- 4.03. getting value #3
-- 4.04. getting value #2
-- 4.05. getting value #1
--
-- 4.06. generating value #1
-- 4.07. generating value #2
-- 4.08. generating value #3
-- 4.09. generating value #4
-- 4.10. generating value #5
-- 4.11. generating value #6
-- where ` `
--     represents a single
--     quote (` `) used an
--     escape character
-- and `.foo` to complete
-- email with alias
-- `VendorContactEmail`

```

```

261  CONCAT (
262      AP1.Vendors.VendorContactFName,
263      ' ',
264      AP1.Vendors.VendorContactLName
265  ) AS ContactName,
266  CONCAT (
267      LEFT(AP1.Vendors.VendorContactFName, 1),
268      AP1.Vendors.VendorContactLName,
269      '@',
270
271
272  REPLACE(
273      REPLACE(
274          REPLACE(
275              REPLACE(
276                  REPLACE(
277                      REPLACE(AP1.Vendors.VendorName, ' ', '')
278
279                      , ' ', '')
280
281                      , ' ', '')
282
283                      , '/' , '')
284
285                      , '&' , '')
286
287                      , ' ', '')
288
289      , '.foo'
290  ) AS VendorContactEmail,
291
292  AP1.Vendors.DefaultTermsID,
293  AP1.Vendors.DefaultAccountNo,
294  -- AP1.ContactUpdates.VendorID AS Expr1,
295  CONCAT (
296      AP1.ContactUpdates.FirstName,
297      ' ',
298      AP1.ContactUpdates.LastName
299  ) AS ContactName,
300
301  CASE
302  WHEN (AP1.ContactUpdates.FirstName <> ''
303       OR AP1.ContactUpdates.FirstName <> ' '
304       OR AP1.ContactUpdates.FirstName IS NOT NULL)--
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

326         REPLACE( -- 4.01. getting value #5
327         REPLACE( -- 4.02. getting value #4
328         REPLACE( -- 4.03. getting value #3
329         REPLACE( -- 4.04. getting value #2
330         REPLACE( -- 4.05. getting value #1
331         REPLACE(AP1.Vendors.VendorName, ' ', '')
332         -- 4.06. generating value #1
333         , ' ', '')
334         , ' ', '')
335         , '/' , '')
336         , '&' , '')
337         , ' ', ''),
338         --
339         -- represents a single
340         -- quote (``) used an
341         -- escape character
342         '.foo' -- and `.foo` to complete
343     )
344 ELSE ''
345 END -- end of CASE clause to
346 AS VendorContactEmail -- make email with alias
347 -- `VendorContactEmail`
348 FROM AP1.Vendors
349 LEFT JOIN AP1.ContactUpdates
350 ON AP1.Vendors.VendorID = AP1.ContactUpdates.VendorID;
351
352
353 /* *****
354 As with previous examples, we can use an alias for each table, which
355 in this case, allows us to present neater code.
356
357 `v` for `AP1.Vendors`
358 `c` for `AP1.ContactUpdates`
359 ***** */
360
361 SELECT v.VendorID,
362        v.VendorName,
363        v.VendorAddress1,
364        v.VendorAddress2,
365        v.VendorCity,
366        v.VendorState,
367        v.VendorZipCode,
368        v.VendorPhone,
369        CONCAT (
370        v.VendorContactFName,
371        ' ',
372        v.VendorContactLName
373        ) AS ContactName,
374        CONCAT (
375        LEFT(v.VendorContactFName, 1),
376        v.VendorContactLName,
377        '@',
378        REPLACE(
379        REPLACE(
380        REPLACE(
381        REPLACE(
382        REPLACE(
383        REPLACE(v.VendorName, ' ', '')
384        , ' ', '')
385        , ' ', '')
386        , '/' , '')
387        , '&' , '')
388        , ' ', ''),
389        '.foo'
390        ) AS VendorContactEmail,

```

```

391 v.DefaultTermsID,
392 v.DefaultAccountNo,
393 -- c.VendorID AS Expr1,
394 CONCAT (
395     c.FirstName,
396     ' ',
397     c.LastName
398 ) AS ContactName,
399 CASE
400     WHEN (
401         c.FirstName <> ''
402         OR c.FirstName <> ' '
403         OR c.FirstName IS NOT NULL
404     )
405     AND (
406         c.LastName <> ''
407         OR c.LastName <> ' '
408         OR c.LastName IS NOT NULL
409     )
410     THEN CONCAT (
411         LEFT(c.FirstName, 1),
412         c.LastName,
413         REPLACE(
414             REPLACE(
415                 REPLACE(
416                     REPLACE(
417                         REPLACE(
418                             REPLACE(v.VendorName, ' ', '')
419                             , ', ' , '' )
420                             , ' / ' , '' )
421                             , '&' , '' )
422                             , ' ' , '' )
423                             , '.foo'
424                         )
425                     )
426                 )
427             )
428         )
429     END AS VendorContactEmail
430 FROM AP1.Vendors AS v
431 LEFT JOIN AP1.ContactUpdates AS c
432     ON v.VendorID = c.VendorID;

```

```

433 /* *****

```

```

434 2. Before you continue learning about SQL
435 (https://searchsqlserver.techtarget.com/definition/SQL) syntax
436 (https://whatis.techtarget.com/definition/syntax), we should cover some
437 important theory, which you will need whether you need to learn SQL to run
438 queries at work and/or you decide to become a database administrator (DBA).
439

```

```

440 2.01. SQL (Structured Query Language) is a standardized programming
441 language used for managing relational databases and performing
442 various operations on the data in them. Initially created in the
443 1970s, SQL is regularly used by database administrators, as well as
444 by developers writing data integration scripts and data analysts
445 looking to set up and run analytical queries.
446 https://searchsqlserver.techtarget.com/definition/SQL
447

```

```

448 2.02. ISO/IEC 9075-1:2016 [SQL:2016] describes the conceptual framework
449 used in other parts of ISO/IEC 9075 to specify the grammar of SQL and
450 the result of processing statements in that language by an
451 SQL-implementation.
452 ISO/IEC 9075-1:2016 also defines terms and notation used in the other
453 parts of ISO/IEC 9075.
454 https://www.iso.org/standard/63555.html
455

```


456 2.03. T-SQL (Transact-SQL) is a set of programming extensions from Sybase
457 and Microsoft that add several features to the Structured Query
458 Language (SQL), including transaction control, exception and error
459 handling, row processing and declared variables.
460 <https://searchsqlserver.techtarget.com/definition/T-SQL>
461

462 2.04. A relational database is a set of tables containing data fitted into
463 predefined categories. Each table (which is sometimes called a
464 relation) contains one or more data categories in columns. Each row
465 contains a unique instance of data for the categories defined by the
466 columns.
467 <http://searchsqlserver.techtarget.com/definition/relational-database>
468

469 2.05. Microsoft SQL Server is a relational database management system, or
470 RDBMS, that supports a wide variety of transaction processing,
471 business intelligence and analytics applications in corporate IT
472 environments. It's one of the three market-leading database
473 technologies, along with Oracle Database and IBM's DB2.
474 Like other RDBMS software, Microsoft SQL Server is built on top of
475 SQL, a standardized programming language that database administrators
476 (DBAs) and other IT professionals use to manage databases and query
477 the data they contain. SQL Server is tied to Transact-SQL (T-SQL),
478 an implementation of SQL from Microsoft that adds a set of
479 proprietary programming extensions to the standard language. The
480 original SQL Server code was developed in the 1980s by the former
481 Sybase Inc., which is now owned by SAP. Sybase initially built the
482 software to run on Unix systems and minicomputer platforms. It,
483 Microsoft and Ashton-Tate Corp., then the leading vendor of PC
484 databases, teamed up to produce the first version of what became
485 Microsoft SQL Server, designed for the OS/2 operating system and
486 released in 1989.
487 <https://searchsqlserver.techtarget.com/definition/SQL-Server>
488

489 2.06. Another form of flat file is one in which table data is gathered in
490 lines of ASCII text with the value from each table cell separated by
491 a comma and each row represented with a new line. This type of flat
492 file is also known as a comma-separated values file (CSV) file.
493 <http://searchsqlserver.techtarget.com/definition/flat-file>
494

495 2.07. A hierarchical database is a design that uses a one-to-many
496 relationship for data elements. Hierarchical database models use a
497 tree structure that links a number of disparate elements to one
498 `owner,` or `parent,` primary record.
499 <https://www.techopedia.com/definition/19782/hierarchical-database>
500

501 2.08. Data Manipulation Language (DML) is the ``vocabulary used to retrieve
502 and work with data... to add, modify, query, or remove data``
503 (<https://msdn.microsoft.com/en-us/library/ff848766.aspx>).
504

505	SELECT	to retrieve records from one or more tables https://techonthenet.com/sql/select.php
506		
507		
508	INSERT	to insert a one or more records into a table https://techonthenet.com/sql/insert.php
509		
510		
511	UPDATE	to update existing records in the tables https://techonthenet.com/sql/update.php
512		
513		
514	DELETE	to delete a one or more records from a table https://techonthenet.com/sql/delete.php
515		
516		
517	MERGE	to insert, update, or delete operations on a target table based on the results of a join with a source table https://msdn.microsoft.com/en-us/library/bb510625.aspx
518		
519		
520		

521 2.09. Data Definition Language (DDL) is the ``vocabulary used to define
522 data structures... to create, alter, or drop data structures``
523 (<https://msdn.microsoft.com/en-us/library/ff848799.aspx>).

524
525 USE to select any existing database in SQL schema [or output
526 from another query]
527 <http://tutorialspoint.com/sql/sql-select-database.htm>

528
529 CREATE to create and define a table [or other database object]
530 https://techonthenet.com/sql/tables/create_table.php

531
532 ALTER to add a column, modify a column, drop a column, rename a
533 column or rename a table [or other database object]
534 https://techonthenet.com/sql/tables/alter_table.php

535
536 DROP to remove or delete a table [or other database object]
537 https://techonthenet.com/sql/tables/drop_table.php

538
539 TRUNCATE to remove all records from a table
540 <https://techonthenet.com/sql/truncate.php>

541
542 DELETE to delete a one or more records from a table
543 <https://techonthenet.com/sql/delete.php>

544
545 2.10. Note that some of these statements can do more than what is covered
546 in these notes for our first sessions.

547
548 For example, the `CREATE` statement is also used to create other
549 database objects as well as access management, but we will not cover
550 these other statements yet. Refer to
551 <https://msdn.microsoft.com/en-us/library/cc879262.aspx> for more
552 information on the `CREATE` statement.

553
554 On a personal note, when looking for information and/or explanation
555 on how to use Microsoft technologies, in this case SQL Server, go to
556 <https://techonthenet.com/> or <http://tutorialspoint.com/> as
557 <https://msdn.microsoft.com/> and other Microsoft websites often seem
558 to be written for advanced users.

559
560 We will use DML and DDL in detail later in the course.

561
562 3. There are several data types
563 (<https://msdn.microsoft.com/en-us/library/ms187752.aspx>) that you need to
564 know if you are interested in taking the certification exam for Database
565 Fundamentals. In everyday use, these are the most often used data types in
566 T-SQL (<http://searchsqlserver.techtarget.com/definition/T-SQL>) -- the
567 version of SQL (<http://searchsqlserver.techtarget.com/definition/SQL>) used
568 in SQL Server (<http://searchsqlserver.techtarget.com/definition/SQL-Server>)
569 -- are the following.

570
571 INT -2³¹ (-2,147,483,648) to 2³¹-1 (2,147,483,647)
572 <https://technet.microsoft.com/en-us/library/ms187745.aspx>

573
574 DECIMAL fixed precision and scale numbers, 10³⁸+1 through 10³⁸-1
575 <https://msdn.microsoft.com/en-us/library/ms187746.aspx>
576 * instead of DOUBLE or FLOAT, indicating the whole value
577 followed by the number of decimals where pi(1,10) can
578 hold 3.1415926536, but not 3.14159265359 for eleven (11)
579 decimal spaces

580
581 VARCHAR(n) 2³¹-1 bytes (2 GB); variable-length, ASCII

582
583 (<http://whatis.techtarget.com/definition/ASCII-American-Standard-Code-for-Information-Interchange>)
string data

584 <https://technet.microsoft.com/en-us/library/ms176089.aspx>
585 not to be confused with NVARCHAR(n) -- variable-length,
586 2^31-1 bytes (2 GB), Unicode
587 (<http://whatis.techtarget.com/definition/Unicode>) string
588 data, not part of most relational database management
589 systems (RDBMS)
590 <https://technet.microsoft.com/en-us/library/ms186939.aspx>
591
592 DATE date
593 <https://technet.microsoft.com/en-us/library/bb630352.aspx>
594
595 TIME time
596 <https://technet.microsoft.com/en-us/library/bb677243.aspx>
597
598 DATETIME defines a date that is combined with a time of day with
599 fractional seconds that is based on a 24-hour clock
600 <https://technet.microsoft.com/en-us/library/ms187819.aspx>
601
602 MONEY money, not part of most relational database management
603 systems (RDBMS)
604 <https://technet.microsoft.com/en-us/library/ms179882.aspx>
605

606 3.01. Conversion may only take place between data similar types.

CONVERSION INPUT	CONVERSION OUTPUT
INT to DECIMAL	no loss; decimal spaces added (.00)
DECIMAL to INT	possible loss of decimal spaces; truncated, value not rounded up or down
DECIMAL to MONEY	truncated and rounded to four decimal spaces for mathematical calculations (.0000 to .9999); two decimal spaces shown for cents (.00 to .99)
DATETIME to DATE	date only; time dropped
DATETIME to TIME	time only; date dropped
DATE to DATETIME	date with default value of `00:00.00.000`
TIME to DATETIME	time with default value of `1900/01/01`
INT DECIMAL DATETIME to VARCHAR DATE NVARCHAR TIME	converted to text; no longer numeric data and cannot be used in mathematical calculations
INT DECIMAL VARCHAR to DATETIME NVARCHAR DATE TIME	straight conversion to proper data type as long as the string field only has numbers and structure is correct (for example, text with value of `2019/03/11` to DATE); no conversion if the string has letters or special characters

649	VARCHAR to NVARCHAR	straight conversion; no data
650		loss
651	+-----+	
652	NVARCHAR to VARCHAR	straight conversion if string is
653		encoded as ACIII or UTF-8;
654		possible data loss if string is
655		encoded as Unicode or no
656		conversion at all
657	+-----+	

3.02. Refer to <https://technet.microsoft.com/en-us/library/ms187912.aspx> for information on approximate numeric data types -- FLOAT and REAL. If you are considering taking the certification, you should know the concept below and why Microsoft recommends not using approximate numeric data types.

```

`The float and real data types are known as approximate
data types. The behavior of float and real follows the
IEEE 754 specification on approximate numeric data types.
Approximate numeric data types do not store the exact
values specified for many numbers; they store an extremely
close approximation of the value. For many applications,
the tiny difference between the specified value and the
stored approximation is not noticeable. At times, though,
the difference becomes noticeable. Because of the
approximate nature of the float and real data types, do not
use these data types when exact numeric behavior is
required, such as in financial applications, in operations
involving rounding, or in equality checks. Instead, use
the integer, decimal, money, or smallmoney data types.
Avoid using float or real columns in WHERE clause search
conditions, especially the = and <> operators. It is best
to limit float and real columns to > or < comparisons. The
IEEE 754 specification provides four rounding modes: round
to nearest, round up, round down, and round to zero.
Microsoft SQL Server uses round up. All are accurate to
the guaranteed precision but can result in slightly
different floating-point values. Because the binary
representation of a floating-point number may use one of
many legal rounding schemes, it is impossible to reliably
quantify a floating-point value.`
https://technet.microsoft.com/en-us/library/ms187912.aspx

```

Note that FLOAT is commonly used in other relational database management systems (RDBMS) like Oracle (<http://oracle.com/>) and in most programming languages including those distributed by Microsoft.

4. As we start, we keep in mind that the most basic structure of a `SELECT` statement (<https://techonthenet.com/sql/select.php>) is the following.

```

SELECT field1, field2...
FROM table1

```

From the previous structure, you can add clauses in the following order. If you organize the clauses any other order, the query will not work.

```

SELECT table1.field1,          -- 1. calling columns/fields
       table1.field2,         --    (data)
       ...
       table2.field1,
       table2.field2,
       ...
       table3.field1,
       table3.field2,
       ...

```

```

714
715         FROM table1                -- 2. where to find data
716                                     --   (tables/views)
717         INNER|LEFT|RIGHT JOIN table2
718             ON table1.shared_field1 = table2.shared_field1
719             AND table1.shared_field2 = table2.shared_field2
720             ...
721         INNER|LEFT|RIGHT JOIN table3
722             ON table1.shared_field1 = table3.shared_field1
723             AND table1.shared_field2 = table3.shared_field2
724             ...
725
726         WHERE condition1             -- 3. filtering output, what
727             AND|OR condition2        --   rows/records you want to
728             AND|OR condition3        --   retrieve
729             ...
730
731         GROUP BY table1.field1,      -- 4. grouping fields not in an
732             table1.field2,          --   aggregate function
733             ...
734             table2.field1,
735             table2.field2,
736             ...
737             table3.field1,
738             table3.field2,
739             ...
740
741         ORDER BY                    -- 5. organizing rows/records
742             table1.field1 ASC|DESC,  --   (output) in ascending
743             table1.field2 ASC|DESC,  --   (`ASC`) or descending
744             ...                      --   (`DESC`) order
745             table2.field1 ASC|DESC,
746             table2.field2 ASC|DESC,
747             ...
748             table3.field1 ASC|DESC,
749             table3.field2 ASC|DESC,
750             ...

```

4.01. In the example below, we retrieve all (`*`) columns from table `AP1.Vendors`.

```

754 ***** */

```

```

756 SELECT *
757 FROM AP1.Vendors;                -- retrieves all values from
758                                     -- table `AP1.Vendors`

```

```

761 /* *****
762     4.02. The only time you can use `SELECT` without `FROM` is when you want
763           the machine to return a value, similar to `PRINT`.
764 ***** */

```

```

766 SELECT 9 * 8;                    -- returns integer 72 (a
767                                     -- mathematical equation)
768
769 SELECT 'Hello there';           -- returns string `Hello there`
770                                     -- (a simple string)

```

```

773 /* *****
774     4.02. As you can see in the examples above, we are not retrieving data from
775           any table. You can get the same results using `PRINT`.
776 ***** */

```

```

778 PRINT 9 * 8;                    -- prints integer 72 (a

```

```

779                                     -- mathematical equation)
780
781 PRINT 'Hello there';                 -- prints string `Hello there`
782                                     -- (a simple string)
783
784
785 /* *****
786 5. We have covered built-in functions that affect strings.
787
788 CONCAT()                             allows you to concatenate strings together
789                                     https://techonthenet.com/sql\_server/functions/concat.php
790
791 + (plus)                             allows you to concatenate 2 or more strings together
792                                     https://techonthenet.com/sql\_server/functions/concat2.php
793
794 LEFT()                               allows you to extract a substring from a string, starting
795 from the left-most character
796                                     https://techonthenet.com/sql\_server/functions/left.php
797
798 LEN()                                returns the length of the specified string... does not
799 include trailing space characters at the end the string
800 when calculating the length
801                                     https://techonthenet.com/sql\_server/functions/len.php
802
803 LTRIM()                              removes all space characters from the left-hand side of a
804 string
805                                     https://techonthenet.com/sql\_server/functions/ltrim.php
806
807 LOWER()                              converts all letters in the specified string to lowercase
808                                     https://techonthenet.com/sql\_server/functions/lower.php
809
810 REPLACE()                            replaces a sequence of characters in a string with another
811 set of characters, not case-sensitive
812                                     https://techonthenet.com/sql\_server/functions/replace.php
813
814 RIGHT()                              allows you to extract a substring from a string, starting
815 from the right-most character
816                                     https://techonthenet.com/sql\_server/functions/right.php
817
818 RTRIM()                              removes all space characters from the right-hand side of a
819 string
820                                     https://techonthenet.com/sql\_server/functions/rtrim.php
821
822 SUBSTRING()                          allows you to extract a substring from a string
823                                     https://techonthenet.com/sql\_server/functions/substring.php
824
825 UPPER()                              converts all letters in the specified string to uppercase
826                                     https://techonthenet.com/sql\_server/functions/upper.php
827
828 Now we will see functions used with numeric values.
829
830 AVG()                                returns the average value of an expression
831                                     https://techonthenet.com/sql\_server/functions/avg.php
832
833 CEILING()                            returns the smallest integer value that is greater than or
834 equal to a number
835                                     https://techonthenet.com/sql\_server/functions/ceiling.php
836
837 COUNT()                              returns the count of an expression
838                                     https://techonthenet.com/sql\_server/functions/count.php
839
840 FLOOR()                              returns the largest integer value that is equal to or less
841 than a number
842                                     https://techonthenet.com/sql\_server/functions/floor.php
843

```

844 MAX() returns the maximum value of an expression
 845 https://techonthenet.com/sql_server/functions/max.php
 846
 847 MIN() returns the minimum value of an expression
 848 https://techonthenet.com/sql_server/functions/min.php
 849
 850 RAND() returns a random number or a random number within a range
 851 https://techonthenet.com/sql_server/functions/rand.php
 852
 853 ROUND() returns a number rounded to a certain number of decimal
 854 places
 855 https://techonthenet.com/sql_server/functions/round.php
 856
 857 SUM() returns the summed value of an expression
 858 https://techonthenet.com/sql_server/functions/sum.php
 859

860 6. In the examples below, we use each one of the numeric functions with the
 861 answer for each on the comment on the right.

862 `***** */`

```

864 SELECT SUM(InvoiceTotal) AS InvoiceTotalSUM, -- returns 214290.51
865        AVG(InvoiceTotal) AS InvoiceTotalAVG, -- returns 1879.7413
866        COUNT(InvoiceTotal) AS InvoiceTotalCOUNT, -- returns 114
867        ROUND(InvoiceTotal, 1) AS InvoiceTotalROUND, -- returns 3813.30
868                                           -- 40.20 ...
869        FLOOR(InvoiceTotal) AS InvoiceTotalFLOOR, -- returns 3813.00
870                                           -- 40.00 ...
871        CEILING(InvoiceTotal) AS InvoiceTotalCEILING, -- returns 3814.00
872                                           -- 41.00 ...
873        MAX(InvoiceTotal) AS InvoiceTotalMAX, -- returns 37966.19
874        MIN(InvoiceTotal) AS InvoiceTotalMIN, -- returns 6.00
875        RAND(InvoiceTotal) AS InvoiceTotalRAND, -- returns 0.713591993212924
876                                           -- 0.713610626184182...
877        FORMAT(InvoiceTotal, 'c', 'en-us') -- `c` for currency with
878        AS InvoiceTotal, -- culture `en-us` (English US)
879                                           -- returns $3,813.33
880                                           -- $40.20 ...
881        FORMAT(InvoiceDueDate, 'd', 'en-us') -- `d` (lower case) for short
882        AS InvoiceDueDate, -- date returning no leading
883                                           -- zeros with culture `en-us`
884                                           -- (English US);
885                                           -- returns 1/8/2012
886                                           -- 1/10/2012 ...
887        FORMAT(InvoiceDueDate, 'D', 'en-us') -- `D` (upper case) for long
888        AS InvoiceDueDate, -- date returning full day of
889                                           -- the week, full month, no
890                                           -- leading zeros with culture
891                                           -- `en-us` (English US);
892                                           -- returns
893                                           -- Sunday, January 8, 2012
894                                           -- Tuesday, January 10, 2012
895                                           -- ...
896        FORMAT(InvoiceDueDate, 'MM/dd/yyyy', 'en-us') -- custom date using format
897        AS InvoiceDueDate -- `MM/dd/yyyy` which overrides
898                                           -- culture `en-us` (English
899                                           -- US); returns 01/08/2012
900                                           -- 01/10/2012 ...

```

```

901 FROM AP1.Invoices
902 GROUP BY InvoiceTotal,
903        AP1.Invoices.InvoiceDueDate
904
905

```

906 `/* ***** */`

907 6.01. When using an aggregate function, we must use `GROUP BY` and list all
 908 columns not in affected by any aggregate function.

909
910 In the example below, we retrieve `VendorState` plus the count of
911 column `VendorState` for each `VendorState` (`COUNT(VendorState)`).

912
913 We can use `DISTINCT` to make sure that duplicate values (rows) are
914 not included in the output of a query.

915
916 We can use `ORDER BY` to organize output by a specific column or list
917 of columns.

918
919 The default option for `ORDER BY` is ascending (`ASC`), which can be
920 omitted (1, 2, 3... a, b, c...).

921
922 The opposite option for `ORDER BY` is descending (`DESC`), which must
923 be used if needed (...3, 2, 1 ...c, b, a).

924 `***** */`

```
925  
926 SELECT DISTINCT -- 1. to avoid duplicates  
927 VendorState, -- 2. column not in aggregate  
928 -- function  
929 COUNT(VendorState) -- 3. column in aggregate  
930 -- function (calculation)  
931 FROM AP1.Vendors -- 4. from table `AP1.Vendors`  
932 GROUP BY VendorState -- 5. must use `GROUP BY` when  
933 -- using any aggregate  
934 -- function, listing all  
935 -- columns not in the  
936 -- aggregate function  
937 ORDER BY VendorState ASC; -- 6. organizing results by  
938 -- column `VendorState` in  
939 -- ascending order
```

940
941
942 `/* *****`

943 6.02. In the example below, we retrieve `VendorID` plus the sum of column
944 `PaymentTotal` for each `VendorID` (`SUM(PaymentTotal)`).

945 `***** */`

```
946  
947 SELECT DISTINCT -- 1. to avoid duplicates  
948 VendorID, -- 2. column not in aggregate  
949 -- function  
950 SUM(PaymentTotal) -- 3. column in aggregate  
951 -- function (calculation)  
952 FROM AP1.Invoices -- 4. from table `AP1.Invoices`  
953 GROUP BY VendorID -- 5. must use `GROUP BY` when  
954 -- using any aggregate  
955 -- function, listing all  
956 -- columns not in the  
957 -- aggregate function  
958 ORDER BY VendorID DESC; -- 6. organizing results by  
959 -- column `VendorID` in  
960 -- descending order
```

961
962
963 `/* *****`

964 7. In the example below, the query returns all values from the `AP1.Vendors`
965 table with all related values from table `AP1.Invoices`,
966 `AP1.InvoiceLineItems` and `AP1.Terms`.

967
968 7.01. The relation between related tables `AP1.Invoices`,
969 `AP1.InvoiceLineItems` and `AP1.Terms` is `INNER JOIN` since the
970 value (row ID) of one table is referenced in another.

971
972 7.02. Dollar amounts are formatted as `c` (currency) with culture `en-us`
973 (English-United States). Dates are formatted as `MM/dd/yyyy` (two

974 digits for month and day, four digits for year) and culture `en-us`
 975 (English-United States). Refer to
 976 <https://msdn.microsoft.com/en-us/library/hh213506.aspx> for more
 977 information. Note that formatting a numeric value changes it to an
 978 alpha-numeric value -- change in data type.
 979

980 7.03. To include the average value of `InvoiceTotal` of all records from
 981 table `AP1.Invoices`, we use a sub-query (also referred to as nested
 982 query, <http://tutorialspoint.com/sql/sql-sub-queries.htm>). We use
 983 alias `AvgInvoiceTotal` to refer to this new column.
 984

```

985     (
986     SELECT FORMAT(AVG(AP1.Invoices.InvoiceTotal),'c','en-us')
987     FROM AP1.Invoices
988     )
989     AS AvgInvoiceTotal
  
```

991 There are various values for culture (one per language and country
 992 combination). The following are just a few, probably the most common
 993 in American businesses. Refer to
 994 <http://sql-server-helper.com/sql-server-2012/format-string-function-culture.aspx>
 995 for a more detailed list of cultures.
 996

CULTURE	LANGUAGE	COUNTRY	RESULT
en-us	English	USA	dollar
en-gb	English	Great Britain	pound
de-de	German	Germany	euro
zh-cn	Simplified Chinese	China	yuan
jp-jp	Japanese	Japan	yen

1012 Refer to <https://www.iso.org/iso-4217-currency-codes.html> for more
 1013 information on currency codes (ISO 4217).
 1014

1015 When formatting DATETIME fields, you can use any of the formats below
 1016 and the culture (`en-us`). The default format in data type DATETIME
 1017 is `yyyy-MM-dd hh:mm:ss.nnnnnn`. Refer to
 1018 <https://docs.microsoft.com/en-us/sql/t-sql/functions/datetime-transact-sql>
 1019 for more information about dates.
 1020

OPTION	OUTPUT	FORMAT
c	currency depending on culture (`\$`)	`c`, `en-us`
d	day without leading zero, day without leading zero and complete year (03/25/2024)	`d`, `en-us`
D	whole day of the week, first letter	`D`, `en-us`

1039		capitalized;	
1040		whole month,	
1041		first letter	
1042		capitalized;	
1043		day without	
1044		leading zero	
1045		and complete	
1046		year (Monday,	
1047		April 17,	
1048		2023)	

DATEPART	OUTPUT	FORMAT
----------	--------	--------

dw	whole day of the week, first letter capitalized (Monday)	`dw MMMM dd, yyyy` `dw MMMM d, yyyy` `dw MMMM dd, yy` `dw MMMM d, yy`
----	--	--

MMMM	whole month, first letter capitalized (April)	`MMMM dd, yyyy` `MMMM d, yyyy` `MMMM dd, yy` `MMMM d, yy`
------	---	--

MMM	month in abbreviation, first letter capitalized (Apr)	`MMM dd, yyyy` `MMM d, yyyy` `MMM dd, yy` `MMM d, yy` `dd-MMM-yy` (default Oracle) `d-MMM-yy` (default Oracle)
-----	---	---

MM	month number with leading zero (04)	`MM/dd/yyyy` `MM/d/yyyy` `MM/dd/yy` `MM/d/yy`
----	-------------------------------------	--

M	month number without leading zero (4)	`M/dd/yyyy` `M/d/yyyy` `M/dd/yy` `M/d/yy`
---	---------------------------------------	--

dddd	day of week (Monday)	`dddd, MMM d, yyyy` `dddd, MMMM d, yyyy`
------	----------------------	---

ddd	day of week abbreviation (Mon)	`ddd, MMM d, yyyy` `ddd, MMMM d, yyyy`
-----	--------------------------------	---

dd	day with leading zero (17)	`MM/dd/yyyy` `M/dd/yyyy` `MM/dd/yy` `M/dd/yy`
----	----------------------------	--

d	day without leading zero (17)	`MM/d/yyyy` `M/d/yyyy` `MM/d/yy` `M/d/yy`
---	-------------------------------	--

yy	last two digits of year (23)	`M/dd/yy` `M/d/yy` `MM/d/yy` `M/d/yy`
----	------------------------------	--

1104	yyyy	complete year (2023)	`M/dd/yyyy`
1105			`M/d/yyyy`
1106			`MM/d/yyyy`
1107			`M/d/yyyy`
1108			
1109	HH	24-hour, military time with leading zero (20)	`HH:mm:ss`
1110			
1111			
1112			
1113			
1114	H	24-hour, military time without leading zero (20)	`H:mm:ss`
1115			
1116			
1117			
1118			
1119			
1120	hh	12-hour (AM/PM), with leading zero (08 PM)	`hh:mm:ss`
1121			
1122			
1123			
1124			
1125	h	12-hour (AM/PM), without leading zero (8 PM)	`h:mm:ss`
1126			
1127			
1128			
1129			
1130			
1131	mm	minutes (13)	`HH:mm:ss`
1132			`H:mm:ss`
1133	ss	seconds (58)	`hh:mm:ss`
1134			`h:mm:ss`
1135			
1136	nnnnnn	six decimal spaces, fractions of a second	`HH:mm:ss.nnnnnn`
1137			`H:mm:ss.nnnnnn`
1138			`hh:mm:ss.nnnnnn`
1139			`h:mm:ss.nnnnnn`
1140			

Although we are using aggregate function `AVG()`, we do not need to use `GROUP BY` since the function is inside the sub-query.

Go to <https://docs.microsoft.com/en-us/sql/t-sql/functions/format-transact-sql> for more information on `FORMAT()`.

***** */

```

1149 SELECT DISTINCT AP1.Vendors.VendorID,
1150 AP1.Vendors.VendorName,
1151 CONCAT ( -- 1. concatenating
1152 AP1.Vendors.VendorAddress1, -- `VendorAddress1`, an
1153 ' ', -- empty space and
1154 AP1.Vendors.VendorAddress2 -- `VendorAddress2`
1155 ) AS VendorAddress, -- as `VendorAddress`
1156 AP1.Vendors.VendorCity,
1157 AP1.Vendors.VendorState,
1158 CONCAT ( -- 2. concatenating
1159 AP1.Vendors.VendorZipCode, -- `VendorZipCode` and a
1160 '-0000' -- dummy Plus4 as
1161 ) AS VendorZipCode, -- VendorZipCode
1162
1163
1164 CASE
1165 WHEN AP1.Vendors.VendorPhone <> '' -- 3. checking that
1166 OR AP1.Vendors.VendorPhone <> ' ' -- `VendorPhone` is not an
1167 OR AP1.Vendors.VendorPhone IS NOT NULL -- empty string or not a
1168 -- space or not `IS NOT

```

```

1169 -- NULL` (in other
1170 -- words, not no-value; must
1171 -- have a value) using
1172 THEN CONCAT ( -- 4. concatenating an opening
1173 '(' , -- parenthesis, the first 3
1174 LEFT(AP1.Vendors.VendorPhone, 3), -- characters of
1175 -- `VendorPhone` (area
1176 -- code), corresponding
1177 ') ', -- closing parenthesis with
1178 SUBSTRING(AP1.Vendors.VendorPhone, 4, 3), -- a space, the substring
1179 -- from `VendorPhone`
1180 -- starting with character 4
1181 -- taking 3 characters
1182 -- (branch exchange), a
1183 '- ', -- hyphen and the 4 four
1184 RIGHT(AP1.Vendors.VendorPhone, 4) -- characters of
1185 -- `VendorPhone`
1186 -- (subscriber number) using
1187 ) -- alias `VendorPhone`
1188 ELSE ''
1189 END
1190 AS VendorPhone,
1191 LTRIM(RTRIM( -- 5. trimming the output of
1192 -- the concatenation of
1193 CONCAT(AP1.Vendors.VendorContactLName, -- `VendorContactLName`, a
1194 ', ', -- comma with a space and
1195 AP1.Vendors.VendorContactFName)) -- `VendorContactFName`
1196 ) AS VendorContactName, -- using alias
1197 -- `VendorContactName`
1198 AP1.Vendors.DefaultAccountNo,
1199 AP1.Invoices.InvoiceID,
1200 AP1.Invoices.InvoiceNumber,
1201 FORMAT(AP1.Invoices.InvoiceDate, -- 5. formatting column as
1202 'MM/dd/yyyy', 'en-us') -- `MM/dd/yyyy` (date) with
1203 -- culture `en-us` as
1204 AS InvoiceDate, -- `InvoiceDate`
1205 FORMAT(AP1.Invoices.InvoiceTotal, -- 7. formatting column as
1206 'c', 'en-us') -- `c` (currency) with
1207 -- culture `en-us` with
1208 AS InvoiceTotal, -- alias `InvoiceTotal`
1209 (
1210 SELECT -- 8. embedded query calling
1211 FORMAT(AVG(AP1.Invoices.InvoiceTotal), -- `AVG(InvoiceTotal)`
1212 'c', 'en-us') -- formatted as `c`
1213 -- (currency) with culture
1214 -- `en-us`
1215 FROM AP1.Invoices -- from all values in table
1216 -- `AP1.Invoices` as
1217 ) AS AvgInvoiceTotal, -- `AvgInvoiceTotal`
1218 FORMAT(AP1.Invoices.PaymentTotal, -- 9. formatting column as `c`
1219 'c', 'en-us') -- (currency) with culture
1220 AS PaymentTotal, -- `en-us` as `PaymentTotal`
1221 FORMAT(AP1.Invoices.CreditTotal, -- 10. formatting column as `c`
1222 'c', 'en-us') -- (currency) with culture
1223 AS CreditTotal, -- `en-us` as `CreditTotal`
1224 FORMAT(AP1.Invoices.InvoiceDueDate, -- 11. formatting column as
1225 'MM/dd/yyyy', 'en-us') -- `MM/dd/yyyy` (date) with
1226 -- culture `en-us` as
1227 AS InvoiceDueDate, -- `InvoiceDueDate`
1228 FORMAT(AP1.Invoices.PaymentDate, -- 12. formatting column as
1229 'MM/dd/yyyy', 'en-us') -- `MM/dd/yyyy` (date) with
1230 -- culture `en-us` as
1231 AS PaymentDate, -- `PaymentDate`
1232 AP1.InvoiceLineItems.InvoiceSequence,
1233 AP1.InvoiceLineItems.AccountNo,

```

```

1234     FORMAT(AP1.InvoiceLineItems.InvoiceLineItemAmount,
1235             -- 13. formatting column as
1236             'c', 'en-us') --      `c` (currency) with
1237                             --      culture `en-us` as
1238     AS InvoiceLineItemAmount, --      `InvoiceLineItemAmount`
1239     AP1.InvoiceLineItems.InvoiceLineItemDescription,
1240     AP1.Terms.TermsDescription,
1241     AP1.Terms.TermsDueDays
1242 FROM AP1.InvoiceLineItems -- 14. from
1243                             --      `AP1.InvoiceLineItems`
1244 INNER JOIN AP1.Invoices --      using `INNER JOIN` to
1245                             --      to connect to
1246                             --      `AP1.Invoices` to get
1247                             --      all shared values from
1248     ON AP1.InvoiceLineItems.InvoiceID = AP1.Invoices.InvoiceID
1249                             --      `AP1.InvoiceLineItems`
1250                             --      and `AP1.Invoices`
1251 INNER JOIN AP1.Terms --      using `INNER JOIN` to
1252                             --      connect to `AP1.Terms`
1253                             --      to get all shared values
1254                             --      from
1255     ON AP1.Invoices.TermsID = AP1.Terms.TermsID --      (`AP1.InvoiceLineItems`
1256                             --      and `AP1.Invoices`) and
1257                             --      `AP1.Terms` using
1258 RIGHT JOIN AP1.Vendors --      `RIGHT JOIN` to connect
1259                             --      to `AP1.Vendors` to get
1260                             --      values from
1261                             --      `AP1.Vendors` and
1262                             --      related data from
1263     ON AP1.Invoices.VendorID=AP1.Vendors.VendorID --      (`AP1.InvoiceLineItems`
1264                             --      and `AP1.Invoices` and
1265                             --      `AP1.Terms`)
1266 ORDER BY -- 15. ordering results by
1267     AP1.Vendors.VendorName, --      `VendorName` first and
1268     AP1.Invoices.InvoiceID; --      then by `InvoiceID`
1269
1270
1271 /* *****
1272     As with previous example, we can use an alias for each table, which
1273     in this case, allows us to present neater code.
1274
1275         `il` for `AP1.InvoiceLineItems`
1276         `i`  for `AP1.Invoices`
1277         `t`  for `AP1.Terms`
1278         `v`  for `AP1.Vendors`
1279 ***** */
1280
1281 SELECT DISTINCT v.VendorID,
1282     v.VendorName,
1283     CONCAT (
1284         v.VendorAddress1,
1285         ' ',
1286         v.VendorAddress2
1287     ) AS VendorAddress,
1288     v.VendorCity,
1289     v.VendorState,
1290     CONCAT (
1291         v.VendorZipCode,
1292         '-0000'
1293     ) AS VendorZipCode,
1294     CASE
1295     WHEN v.VendorPhone <> ''
1296     OR v.VendorPhone <> ' '
1297     OR v.VendorPhone IS NOT NULL
1298     THEN CONCAT (

```

```

1299         '(',
1300         LEFT(v.VendorPhone, 3),
1301         ') ',
1302         SUBSTRING(v.VendorPhone, 4, 3),
1303         RIGHT(v.VendorPhone, 4)
1304     )
1305 ELSE ''
1306 END AS VendorPhone,
1307 LTRIM(RTRIM(CONCAT (
1308     v.VendorContactLName,
1309     ', ',
1310     v.VendorContactFName
1311 ))) AS VendorContactName,
1312 v.DefaultAccountNo,
1313 i.InvoiceID,
1314 i.InvoiceNumber,
1315 FORMAT(i.InvoiceDate, 'MM/dd/yyyy', 'en-us') AS InvoiceDate,
1316 FORMAT(i.InvoiceTotal, 'c', 'en-us') AS InvoiceTotal,
1317 (
1318     SELECT FORMAT(AVG(i.InvoiceTotal), 'c', 'en-us')
1319     FROM AP1.Invoices AS i
1320                                     -- 1. alias `i` not needed as
1321                                     --    this is a sub-query, but
1322                                     --    solely included for
1323                                     --    aesthetics in the example
1324                                     -- 2. outer aliases needed
1325 ) AS AvgInvoiceTotal,
1326 FORMAT(i.PaymentTotal, 'c', 'en-us') AS PaymentTotal,
1327 FORMAT(i.CreditTotal, 'c', 'en-us') AS CreditTotal,
1328 FORMAT(i.InvoiceDueDate, 'MM/dd/yyyy', 'en-us') AS InvoiceDueDate,
1329 FORMAT(i.PaymentDate, 'MM/dd/yyyy', 'en-us') AS PaymentDate,
1330 il.InvoiceSequence,
1331 il.AccountNo,
1332 FORMAT(il.InvoiceLineItemAmount, 'c', 'en-us')
1333     AS InvoiceLineItemAmount,
1334 il.InvoiceLineItemDescription,
1335 t.TermsDescription,
1336 t.TermsDueDays
1337 FROM AP1.InvoiceLineItems AS il
1338 INNER JOIN AP1.Invoices AS i
1339     ON il.InvoiceID = i.InvoiceID
1340 INNER JOIN AP1.Terms AS t
1341     ON i.TermsID = t.TermsID
1342 RIGHT JOIN AP1.Vendors AS v
1343     ON i.VendorID = v.VendorID
1344 ORDER BY v.VendorName,
1345     i.InvoiceID;
1346
1347 /* *****
1348 8. To get the difference between two dates, we use `DATEDIFF()`, which
1349 ``returns the difference between two date values, based on the interval
1350 specified`` (https://techonthenet.com/sql\_server/functions/datediff.php).
1351
1352 We also call functions `DAY()`,
1353 (https://techonthenet.com/sql\_server/functions/day.php), `MONTH()`,
1354 (https://techonthenet.com/sql\_server/functions/month.php) and `YEAR()`,
1355 (https://techonthenet.com/sql\_server/functions/year.php).
1356
1357 8.01. In the example below, we use `01/01/2017` as the starting date and
1358 `03/25/2024` as the end date.
1359 ***** */
1360
1361 SELECT DATEDIFF(DAY, '01/01/2017', '03/25/2024') AS DatediffDays, -- 2,640 days
1362     DATEDIFF(MONTH, '01/01/2017', '03/25/2024') AS DatediffMonths, -- 86 months
1363     DATEDIFF(YEAR, '01/01/2017', '03/25/2024') AS DatediffYears; -- 7 years

```

```
1364
1365
1366 /* *****
1367     8.02. Instead of hard-coding today's date, we can use function `GETDATE()`
1368     to retrieve the local system datetime -- `2024-03-25 21:52:16.327`.
1369     ***** */
1370
1371 SELECT DATEDIFF(DAY, '01/01/2017', GETDATE()) AS DatediffDays, -- 2,640 days
1372        DATEDIFF(MONTH, '01/01/2017', GETDATE()) AS DatediffMonths, -- 86 months
1373        DATEDIFF(YEAR, '01/01/2017', GETDATE()) AS DatediffYears; -- 7 years
1374
1375
1376 /* *****
1377 https://folvera.commons.gc.cuny.edu/?p=1277
1378     ***** */
```