

```

1          /*      DATABASE CREATION      */
2  CREATE DATABASE PROJECT2;
3
4  USE PROJECT2;
5
6  --DROP DATABASE PROJECT2
7          /*      TABLE CREATION      */
8
9  CREATE TABLE CATEGORIES
10 (
11  CID INT PRIMARY KEY,
12  CNAME VARCHAR(33)
13 );
14
15 CREATE TABLE PRODUCTS
16 (
17  PID INT,
18  PNAME VARCHAR(55),
19  PRICE INTEGER,
20  CID INT FOREIGN KEY REFERENCES CATEGORIES(CID)
21 );
22 -- DROP TABLE PRODUCTS
23
24          /*      DATA INSERTION      */
25          */
26 INSERT INTO CATEGORIES VALUES(1, 'FRUITS');
27 INSERT INTO CATEGORIES VALUES(2, 'VEGITABLES');
28 INSERT INTO CATEGORIES VALUES(3, 'SHAMPOOS');
29 INSERT INTO CATEGORIES VALUES(4, 'CONDITIONORS');
30 INSERT INTO CATEGORIES VALUES(5, 'GEL');
31
32 INSERT INTO PRODUCTS VALUES(111, 'grapes', 23, 1)
33 INSERT INTO PRODUCTS VALUES(112, 'onion', 44, 2)
34 INSERT INTO PRODUCTS VALUES(113, 'all clear', 55, 4)
35 INSERT INTO PRODUCTS VALUES(114, 'livion', 66, 5)
36 INSERT INTO PRODUCTS VALUES(115, 'mango', 23, 1)
37 INSERT INTO PRODUCTS VALUES(116, 'SUNSILK', 87, 3)
38 INSERT INTO PRODUCTS VALUES(117, 'LOREAL', 55, 5)
39 INSERT INTO PRODUCTS VALUES(118, 'GILLETE', 66, 5)
40 INSERT INTO PRODUCTS VALUES(119, 'APPLE', 76, 1)
41 INSERT INTO PRODUCTS VALUES(120, 'BLUNT', 54, 3)
42 INSERT INTO PRODUCTS VALUES(121, 'BRINJAL', 99, 2)
43 INSERT INTO PRODUCTS VALUES(122, 'BANANA', 43, 1)
44
45
46 SELECT * FROM CATEGORIES
47 SELECT * FROM PRODUCTS
48          /*      QUERIES      */
49
50 /* 1 LIST ALL THE PRODUCTS THAT BELONGS TO VEGITABLES AND CONDITIONERS*/
51 SELECT PNAME, CNAME FROM PRODUCTS, CATEGORIES
52 WHERE CNAME IN ('VEGITABLES', 'CONDITIONORS') AND PRODUCTS.CID =

```

```

    CATEGORIES.CID
53
54 /* 1. ANOTHER METHOD*/
55 SELECT PNAME FROM PRODUCTS
56 WHERE CID IN (SELECT CID FROM CATEGORIES
57 WHERE CNAME IN ('VEGITABLES', 'CONDITIONORS'))
58
59 /* 2 LIST ALL THE PRODUCTS THAT HAD PRICE RANGE 30 - 65*/
60 SELECT PNAME FROM PRODUCTS
61 WHERE PRICE>=30 AND PRICE<65
62
63 /* 2 ANOTHER METHOD */
64 SELECT PNAME FROM PRODUCTS
65 WHERE PRICE BETWEEN 30 AND 65
66
67 -- FOR UNDERSTANDING
68
69 SELECT PNAME, PRICE FROM PRODUCTS
70 WHERE PRICE BETWEEN 30 AND 65
71
72
73 /*3 SUM OF TOTAL PRICES FOR EACH CATEGORY*/
74
75 SELECT CNAME,SUM(PRICE) AS SUM_PRICES
76 FROM PRODUCTS AS P, CATEGORIES AS C
77 WHERE C.CID=P.CID GROUP BY CNAME
78
79 /* 3. ANOTHER METHOD*/
80 SELECT SUM(P.PRICE) AS CATEGORY_PRICE,C.CNAME FROM
81 CATEGORIES AS C JOIN PRODUCTS AS P ON C.CID = P.CID
82 GROUP BY C.CNAME
83
84 /* 4. PRODUCTS WITH HIGHEST PRICES (5 PRICES) */
85
86 SELECT TOP 5 PRICE FROM PRODUCTS ORDER BY PRICE DESC
87
88 SELECT PNAME FROM PRODUCTS ORDER BY PRICE DESC
89 /*4 ANOTHER METHOD*/
90
91 SELECT TOP 5 PNAME, PRICE FROM PRODUCTS
92 ORDER BY PRICE DESC
93
94 /* 5 ARRANGE PRODUCTS DESCENDING ORDER OF PRICE */
95
96 SELECT * FROM PRODUCTS ORDER BY PRICE DESC
97
98
99
100
101
102 select * from CATEGORIES

```

```
103
104 select * from PRODUCTS
105
106                                     /*    DISTINCT    ↗
                                     */
107
108 select distinct cid from products
109
110 select distinct cid,pname from products
111
112                                     /*    WHERE    ↗
                                     */
113
114 select pname, price from products where cid = 1
115
116 select pname, price from products where price > 50
117
118                                     /*    BETWEEN    ↗
                                     */
119
120 select pname, price from PRODUCTS where price between 40 and 80
121
122                                     /*    AND OR NOT    ↗
                                     */
123
124 select pname,price from products where cid = 1 and pname = 'grapes'
125
126 select pname, cid from products where cid = 1 or cid = 3
127
128 select pname, cid from products where not cid = 1
129
130 select pname, price from products where cid = 2 and (pname = 'onion' or
    pname = 'BRINJAL')
131
132                                     /*    ORDER    ↗
    BY    */
133
134 /* ascending order by default */
135
136 select pname,price from products order by price
137
138 /* descending order */
139
140 select pname, price from products order by price desc
141
142 select pname, cid from products order by cid
143
144 select pname, cid from products order by cid,pname
145 -- previously pname is coming by insertion order
146 -- now by including pname for order by pname getting sorted
147
148 select pname from PRODUCTS order by pname
149
```

```
150 select pname, cid from products order by cid ,pname desc
151 -- one column ascending while another descending order
152
153 select pname, cid from products order by cid asc,pname desc
154
155 select pname, cid from products order by cid desc,pname asc
156
157
158
159 INSERT INTO PRODUCTS (PID,PNAME) VALUES (123,'LIMKA')
160 -- FOR CID AND PRICE NULL
161
162 SELECT * FROM PRODUCTS
163
164 SELECT * FROM PRODUCTS WHERE CID IS NOT NULL
165 -- RESULTS WHICH ARE NOT HAVING NULL
166
167 SELECT * FROM PRODUCTS WHERE CID IS NULL
168 -- RESULTS WHICH ARE HAVING NULL
169
170
171 SELECT * FROM PRODUCTS WHERE CID = 2
172
173 UPDATE PRODUCTS
174 SET CID = 2
175 WHERE PID=123
176
177 SELECT * FROM PRODUCTS WHERE CID = 2
178
179 SELECT * FROM PRODUCTS
180
181 --UPDATE PRODUCTS
182 --SET CID = 2
183
184 SELECT * FROM PRODUCTS
185
186 /* If you omit the WHERE clause,
187 all records in the table will be updated! */
188
189 ALTER TABLE PRODUCTS
190 ADD VALIDITY INT
191
192 SELECT * FROM PRODUCTS
193
194 UPDATE PRODUCTS
195 SET VALIDITY = 10
196 WHERE PID<115
197 -- ONLY SPECIFIED ROWS ARE UPDATED
198
199 UPDATE PRODUCTS
200 SET VALIDITY = 20
```

```
201 -- ALL ROWS ARE UPDATED BECAUSE IT IS NOT MENTIONED WITH WEHRE CLAUSE
202
203 SELECT * FROM PRODUCTS
204
205                                     /* ↻
                                     DELETE */
206
207 DELETE FROM PRODUCTS
208 WHERE PID = 123
209
210 SELECT * FROM PRODUCTS
211
212 /* If you omit the WHERE clause,
213 all records in the table will be deleted! */
214
215 --TO DELETE TABLE
216 /* DELETE FROM TABLE_NAME */
217
218                                     /* ↻
                                     SELECT TOP CLAUSE */
219
220 SELECT * FROM CATEGORIES
221 SELECT * FROM PRODUCTS
222
223 SELECT TOP 5 * FROM PRODUCTS
224
225 SELECT TOP 5 PRICE, PNAME FROM PRODUCTS
226
227 --FOR PRODUCT PRICES IN DESCENDING ORDER(HIGH - LOW)
228 SELECT TOP 5 PRICE,PNAME FROM PRODUCTS ORDER BY PRICE DESC
229
230 --FOR PRODUCT PRICES IN ASCENDING ORDER(LOW - HIGH)
231 SELECT TOP 5 PRICE, PNAME FROM PRODUCTS ORDER BY PRICE
232
233 -- GROUP BY JUST PRACTICE
234 SELECT SUM(PRICE) FROM PRODUCTS GROUP BY CID
235
236                                     ↻
                                     /* MIN */
237
238 --MIN PRICE VALUES IN PRODUCTS
239 SELECT MIN(PRICE) FROM PRODUCTS
240 -- MIN PRICE IN EACH CATEGORY
241 SELECT MIN(PRICE) AS MIN_PRICE,CID FROM PRODUCTS GROUP BY CID
242
243 -- PRODUCT NAME OF MIN PRICE
244 SELECT PNAME,PRICE FROM PRODUCTS WHERE PRICE
245 IN
246 (SELECT MIN(PRICE) AS PRICE FROM PRODUCTS GROUP BY CID)
247
248 -- TOTAL PRICE OF EACH CATEGORY
249 SELECT SUM(PRICE) AS TOTAL_PRICE, CID FROM PRODUCTS GROUP BY CID
250
```

```
251
252          /*      MAX      */
253 -- MAX PRICE AMONG ALL PRODUCTS
254 SELECT MAX(PRICE) FROM PRODUCTS
255
256 -- MAX PRICE IN CATEGORY 1
257 SELECT MAX(PRICE) FROM PRODUCTS WHERE CID =1
258
259 -- MAX PRICE IN EACH CATEGORY
260 SELECT MAX(PRICE),CID FROM PRODUCTS GROUP BY CID
261
262          /*      COUNT      */
263 --TOTAL NO.OF COLUMNS
264 SELECT COUNT(CID) FROM PRODUCTS
265
266 -- NO OF PRODUCTS FOR EACH CATEGORY
267 SELECT COUNT(*) AS COUNTING,CID FROM PRODUCTS GROUP BY CID
268
269 -- NO OF PRODUCTS WHOOSE PRICE IS ABOVE 100
270 SELECT PNAME,PRICE FROM PRODUCTS WHERE PRICE >= 75
271
272 SELECT COUNT(*)FROM PRODUCTS WHERE PRICE >= 75
273
274          /*      AVG      */
275 --AVERAGE PRICES IN PRODUCTS
276 SELECT AVG(PRICE) FROM PRODUCTS
277
278 --AVERAGE PRICES IN PRODUCTS BASED ON CIDS
279 SELECT AVG(PRICE), CID FROM PRODUCTS GROUP BY CID
280
281          /*      SUM      */
282 -- SUM OF ALL PRODUCTS PRICES
283 SELECT SUM(PRICE) FROM PRODUCTS
284
285 -- SUM OF ALL THE PRICES WITH RESPECT TO CID
286 SELECT SUM(PRICE), CID FROM PRODUCTS GROUP BY CID
287
288          /*      LIKE      */
289 SELECT * FROM PRODUCTS
290
291 -- STARTS WITH A
292 SELECT * FROM PRODUCTS WHERE PNAME LIKE 'A%'
293
294 -- ENDS WITH A
295 SELECT * FROM PRODUCTS WHERE PNAME LIKE '%A'
296
297 --ANY IN MIDDLE
298 SELECT * FROM PRODUCTS WHERE PNAME LIKE '%A%'
```

```

299 SELECT * FROM PRODUCTS WHERE PNAME LIKE '%N%'
300
301 -- 2ND POSITION A
302 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '_A%'
303
304 -- 7TH POSITION A
305 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '_____A%'
306
307 --STARTS WITH A AND ATLEAST 3 CHARACTER LENGTH
308 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE 'A__%'
309
310 --STARTS WITH A AND ENDS WITH R
311 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE 'A%R'
312
313
314 /*          NOT LIKE          */
315 -- SHOULD NOT STARTS WITH A
316 SELECT PNAME FROM PRODUCTS WHERE PNAME NOT LIKE 'A%' --ALL CLEAR AND APLLE
317 LEFT
318
319 -- A SHOULD NOT BE IN MIDDLE
320 SELECT PNAME FROM PRODUCTS WHERE PNAME NOT LIKE '%A%'
321
322 /*          wildcards          */
323 /* A wildcard character is used to substitute
324 one or more characters in a string*/
325 --> % _ [] ^ -
326 -- %
327 -- PNAME STARTS WITH A
328 SELECT * FROM PRODUCTS WHERE PNAME LIKE 'A%'
329 -- ALL PNAME CONTAIN AN
330 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '%AN%'
331 -- AFTER 4---- ETE STRING
332 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '____ETE'
333 -- STARTS WITH ANY LETTER SECOND LETTER IS A
334 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '_A%'
335
336 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '_A_A_A'
337
338 --starting with "b", "s", or "p"
339 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '[BSP]%'
340
341 --ENDS with "b", "s", or "p"
342 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '%[BSP]%'
343
344 --starting with "a", "b", or "c":
345 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '[A-C]%'
346
347 -- NOT starting with "b", "s", or "p":
348 SELECT PNAME FROM PRODUCTS WHERE PNAME LIKE '[^AB]%'
349
350 --OR
351 SELECT PNAME FROM PRODUCTS WHERE PNAME NOT LIKE '[AB]%'
352
353

```

```
349          /*          IN          */
350  -- MIN PRICE PRODUCTS
351  SELECT PRICE, PNAME FROM PRODUCTS WHERE PRICE IN(
352  SELECT MIN(PRICE) FROM PRODUCTS)
353
354  SELECT * FROM PRODUCTS WHERE CID IN (1,2,3)
355
356  SELECT * FROM PRODUCTS WHERE CID NOT IN (1,2,3)
357
358          /*          BETWEEN AND NOT BETWEEN          */
359
360  SELECT * FROM PRODUCTS WHERE CID BETWEEN 2 AND 4
361
362  SELECT * FROM PRODUCTS WHERE CID NOT BETWEEN 2 AND 4
363
364
365  SELECT * FROM PRODUCTS WHERE PRICE BETWEEN 51 AND 99 AND CID NOT IN(2,4)
366  -- 99 NOT INCLUDED
367
368  SELECT * FROM PRODUCTS
369  WHERE PNAME BETWEEN 'A' AND 'G' -- G NOT INCLUDED
370
371  SELECT * FROM PRODUCTS
372  WHERE PNAME BETWEEN 'A' AND 'H'
373
374
375  SELECT * FROM PRODUCTS
376  WHERE PNAME BETWEEN 'A' AND 'H' ORDER BY PNAME
377
378  --CAN ALSO USED FOR DATES
379
380          /*          ALIAS          */
381
382  SELECT CID AS CUSTOMERID FROM PRODUCTS
383
384  SELECT DISTINCT CID AS UNIQUECUSTOMER FROM PRODUCTS
385
386  SELECT MAX(PRICE) AS MAXPRICE FROM PRODUCTS
387
388  SELECT P.PID,P.PNAME FROM PRODUCTS AS P
389
390          /*          JOIN          */
391  /* A JOIN clause is used to combine rows from two or more tables,
392  based on a related column between them.*/
393
394  SELECT * FROM PRODUCTS
395  SELECT * FROM CATEGORIES
396
397  SELECT * FROM PRODUCTS AS P, CATEGORIES AS C WHERE P.CID = C.CID
```



```

398 --ABOVE BELOW SAME
399 SELECT * FROM PRODUCTS JOIN CATEGORIES ON PRODUCTS.CID = CATEGORIES.CID
400
401 /* (INNER) JOIN: Returns records that have matching values in both tables
402    LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
403    RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
404    FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table
405 */
406 -----> INNER JOIN
407 SELECT * FROM CATEGORIES AS C INNER JOIN PRODUCTS AS P ON C.CID = P.CID
408 -----> LEFT JOIN
409 SELECT * FROM CATEGORIES AS C LEFT JOIN PRODUCTS AS P ON C.CID = P.CID
410 -----> RIGHT JOIN
411 SELECT * FROM CATEGORIES AS C RIGHT JOIN PRODUCTS AS P ON C.CID = P.CID
412 -----> FULL OUTER JOIN
413 SELECT * FROM CATEGORIES AS C FULL OUTER JOIN PRODUCTS AS P ON C.CID = P.CID
414
415 SELECT * FROM CATEGORIES
416 SELECT * FROM PRODUCTS
417
418             /*      UNION      */
419
420 SELECT CID FROM PRODUCTS
421 UNION
422 SELECT CID FROM CATEGORIES
423 --UNION selects only distinct values.
424
425 --UNION ALL
426 SELECT CID FROM PRODUCTS
427 UNION ALL
428 SELECT CID FROM CATEGORIES
429
430 -- UNION WHERE
431 SELECT CID FROM PRODUCTS WHERE CID>3
432 UNION
433 SELECT CID FROM CATEGORIES WHERE CID>3
434
435 -- UNION ALL WHERE
436 SELECT CID FROM PRODUCTS WHERE CID>3
437 UNION ALL
438 SELECT CID FROM CATEGORIES WHERE CID>3
439
440 SELECT * FROM PRODUCTS WHERE CID IN(
441 SELECT CID FROM PRODUCTS WHERE CID>3
442 UNION ALL
443 SELECT CID FROM CATEGORIES WHERE CID>3)
444
445

```

```
/*      GROUP BY      */
446 SELECT * FROM CATEGORIES
447 SELECT * FROM PRODUCTS
448
449 SELECT CID FROM PRODUCTS
450 GROUP BY (CID)
451 ORDER BY(CID)
452
453 SELECT COUNT(CID),CID FROM PRODUCTS
454 GROUP BY CID
455
456 SELECT COUNT(CID) COUNT_CID,CID FROM PRODUCTS
457 GROUP BY CID
458 ORDER BY COUNT_CID DESC
459
460 SELECT SUM(PRICE) FROM PRODUCTS WHERE CID IN(
461 SELECT CID FROM PRODUCTS
462 GROUP BY(CID))
463
464
465 -- USING SPECIFIC CONDITION CID > 3
466 SELECT COUNT(CID) AS COUNTING,CID FROM PRODUCTS
467 GROUP BY CID
468 HAVING CID>3
469
470 -- CONDITION OF CID BETWEEN 1 AND 4
471 SELECT SUM(PRICE) SUM ,CID FROM PRODUCTS
472 GROUP BY CID
473 HAVING CID BETWEEN 1 AND 4
474
475
476 /*      EXISTS      */
476 SELECT PNAME, PRICE FROM PRODUCTS WHERE EXISTS(
477 SELECT CID FROM PRODUCTS)
478
479 -- DIFFERENCE BETWEEN IN AND EXISTS
480
481 -- EXISTS
482 /* -> The EXISTS operator is used to check the existence of rows returned
    by a subquery.
483 -> It evaluates the subquery and returns true if the subquery returns
    at least one row;
484 -> otherwise, it returns false.
485 */
486
487 -- IN
488 /* ->The IN operator is used to compare a value against a list of values
    or a subquery.
489 -> It checks if a value matches any value in the specified list or the
    result set of a subquery.
490 */
491 SELECT CID FROM PRODUCTS WHERE CID > 3
```

```
492
493 SELECT PNAME, PRICE,CID FROM PRODUCTS WHERE EXISTS(
494 SELECT CID FROM PRODUCTS WHERE CID > 3)
495
496 SELECT PNAME, PRICE,CID FROM PRODUCTS WHERE CID IN (
497 SELECT CID FROM PRODUCTS WHERE CID > 3)
498
499 SELECT CID FROM PRODUCTS WHERE CID BETWEEN 1 AND 4
500
501 SELECT PNAME,PRICE,CID FROM PRODUCTS WHERE EXISTS(
502 SELECT CID FROM PRODUCTS WHERE CID BETWEEN 1 AND 4
503 )
504
505 SELECT PNAME,PRICE,CID FROM PRODUCTS WHERE CID IN(
506 SELECT CID FROM PRODUCTS WHERE CID BETWEEN 1 AND 4
507 )
508
509 ----> EXISTS ONLY ACTS AS A CONDITIONER
510 ----> IF TRUE OUTER QUERY WILL BE EXICUTED
511 ----> IF NOT IT WON'T EXICUTED
512
513
514
515
516 SELECT CID FROM PRODUCTS
517 WHERE CID > 4
518
519 SELECT * FROM PRODUCTS
520 WHERE CID = ANY(
521 SELECT CID FROM PRODUCTS
522 WHERE CID > 4)
523
524
525 SELECT CID FROM PRODUCTS
526 WHERE CID > 5
527
528 SELECT * FROM PRODUCTS
529 WHERE CID = ANY(
530 SELECT CID FROM PRODUCTS
531 WHERE CID > 5)
532
533 SELECT CID FROM PRODUCTS
534 WHERE CID BETWEEN 1 AND 6
535
536 SELECT * FROM PRODUCTS
537 WHERE CID = ALL(
538 SELECT CID FROM PRODUCTS
539 WHERE CID = 1)
540
541 SELECT * FROM PRODUCTS
542 WHERE CID = ALL(
543 SELECT CID FROM PRODUCTS
```



```
544 WHERE CID BETWEEN 1 AND 6)
545
546
547
548                                     ↗
          /*      SELECT INTO      */
549
550
551 /*The SELECT INTO statement copies data from one table into a new table.*/
552
553 --SYNTAX
554
555 /*
556     SELECT column1, column2, column3, ...
557     INTO newtable [IN externaldb]
558     FROM oldtable
559     WHERE condition;
560 */
561 SELECT * INTO PRODUCTS2
562 FROM PRODUCTS
563
564 SELECT * FROM PRODUCTS2
565
566 SELECT PNAME, PRICE INTO PRODUCTS3 FROM PRODUCTS
567
568 SELECT * FROM PRODUCTS3
569
570                                     ↗
          /*      INSERT INTO SELECT      */
571
572
573 /*The INSERT INTO SELECT statement copies data from one table and inserts ↗
   it into another table.
574
575 The INSERT INTO SELECT statement requires that the data types in source ↗
   and target tables match.
576
577 Note: The existing records in the target table are unaffected.
578 TABLE NEED TO BE EXISTS PREVIOUSLY
579
580 */
581 CREATE TABLE PRO (PNAME VARCHAR(25), PRICE INT)
582 INSERT INTO PRO SELECT * FROM PRODUCTS3
583
584 SELECT * FROM PRO
585
586 /*
587 The CASE expression goes through conditions and returns a value when the ↗
   first condition is met (like an if-then-else statement).
588
589 So, once a condition is true, it will stop reading and return the result.
590
591 If no conditions are true, it returns the value in the ELSE clause.
```

```
592
593 If there is no ELSE part and no conditions are true, it returns NULL.
594
595 CASE
596     WHEN condition1 THEN result1
597     WHEN condition2 THEN result2
598     WHEN conditionN THEN resultN
599     ELSE result
600 END;
601
602 */
603
604 SELECT PNAME, CASE
605     WHEN PRICE >=20 AND PRICE <50 THEN 'CHEAP'
606     WHEN PRICE >=50 AND PRICE <75 THEN 'VALUE FOR MONEY'
607     WHEN PRICE >=75 AND PRICE <100 THEN 'EXPENSIVE'
608 ELSE 'UN LABELED'
609 END AS REVIEW
610 FROM PRO;
611
612
613
614
```