

```
1  /*-----*/  
2  
3  /*-----KOMAL MORE - APR BATCH-----*/  
4  
5  /*-----hr - DATABASE EXERCISE-----*/  
6  
7 • USE hr;  
8  
9  /* 1) Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name" */  
10  
11 • SELECT first_name AS "First Name", last_name AS "Last Name"  
12   FROM Employees;  
13  
14  /* 2) Write a query to get unique department ID from employee table */  
15  
16 • SELECT DISTINCT department_id  
17   FROM employees;  
18  
19  /* 3) Write a query to get all employee details from the employee table order by first name, descending */  
20  
  
Output :  
Action Output  
# Time Action Message Duration / Fetch  
109 09:37:50 USEhr 0 row(s) affected 0.000 sec
```

```
1  /*-----SQL PROJECT - 1-----*/
2
3  /*-----KOMAL MORE - APR BATCH-----*/
4
5  /*-----hr - DATABASE EXERCISE-----*/
6
7 • USE hr;
8
9  /* 1) Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name" */
10
11 • SELECT first_name AS "First Name", last_name AS "Last Name"
12   FROM Employees;
13
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

First Name	Last Name
Steven	King
Neena	Kochhar
Lex	De Haan
Alexander	Hunold
Bruce	Ernst
David	Austin

Employees 2 ×

Result Grid

Form Editor

Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
2	09:51:12	SELECT first_name AS "First Name", last_name AS "Last Name" FROM Employees LIMIT 0, ...	107 row(s) returned	0.000 sec / 0.000 sec

```
6
7 • USE hr;
8
9 /* 1) Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name" */
10
11 • SELECT first_name AS "First Name", last_name AS "Last Name"
12 FROM Employees;
13
14 /* 2) Write a query to get unique department ID from employee table */
15
16 • SELECT DISTINCT department_id
17 FROM employees;
18
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

department_id
HULL
10
20
30
40
50
..

employees 3 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
2	09:56:08	SELECT DISTINCT department_id FROM employees LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec

```

13
14      /* 2) Write a query to get unique department ID from employee table */
15
16 •   SELECT DISTINCT department_id
17     FROM employees;
18
19      /* 3) Write a query to get all employee details from the employee table order by first name, descending */
20
21 •   SELECT *
22     FROM employees
23     ORDER BY first_name DESC;
24
25      /* 4) Write a query to get the names (first name, last name), salary, PF of all the employees (PF is calculated as 15% of salary) */

```

**Result Grid** | Filter Rows:  | Edit: Export/Import: | Wrap Cell Content:

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
▶	180	Winston	Taylor	WTAYLOR@example.com	650.507.9876	1998-01-24	SH_CLERK	3200.00	NULL	120	50
	171	William	Smith	WSMITH@example.com	011.44.1343.629268	1999-02-23	SA_REP	7400.00	0.15	148	80
	206	William	Gietz	WGIETZ@example.com	51hr5.123.8181	1994-06-07	AC_ACCOUNT	8300.00	NULL	205	110
	195	Vance	Jones	VJONES@example.com	650.501.4876	1999-03-17	SH_CLERK	2800.00	NULL	123	50
	106	Valli	Pataballa	VPATABAL@example.com	590.423.4560	1998-02-05	IT_PROG	4800.00	NULL	103	60
	141	Trenna	Rajs	TRAJS@example.com	650.121.8009	1995-10-17	ST_CLERK	3500.00	NULL	124	50
	108	Tara	Smith	TMILLIGAN@example.com	555.555.5554	1998-01-12	ST_CLERK	2400.00	NULL	121	50

employees 9 ×

**Output**

**Action Output**

#	Time	Action	Message	Duration / Fetch
3	09:57:39	SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
17     FROM employees;  
18  
19     /* 3) Write a query to get all employee details from the employee table order by first name, descending */  
20  
21 •  SELECT *  
22     FROM employees  
23     ORDER BY first_name DESC;  
24  
25     /* 4) Write a query to get the names (first_name, last_name), salary, PF of all the employees (PF is calculated as 15% of salary) */  
26  
27 •  SELECT CONCAT(first_name, ' ', last_name) AS "names", salary, (0.15*salary) AS PF  
28     FROM employees;  
29
```

Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

	names	salary	PF
▶	Steven King	24000.00	3600.0000
▶	Neena Kochhar	17000.00	2550.0000
▶	Lex De Haan	17000.00	2550.0000
▶	Alexander Hunold	9000.00	1350.0000
▶	Bruce Ernst	6000.00	900.0000
▶	David Austin	4800.00	720.0000
...	...	...	...

Result 10 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
4	14:33:00	SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000	107 row(s) returned	0.047 sec / 0.000 sec

```
23 ORDER BY first_name DESC;  
24  
25 /* 4) Write a query to get the names (first_name, last_name), salary, PF of all the employees (PF is calculated as 15% of salary) */  
26  
27 • SELECT CONCAT(first_name, ' ', last_name) AS "names", salary, (0.15*salary) AS PF  
28 FROM employees;  
29  
30 /* 5) Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary */  
31  
32 • SELECT employee_id, CONCAT(first_name, ' ', last_name) AS "Names", salary  
FROM employees  
ORDER BY salary;  
33  
34  
35
```

Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

employee_id	Names	salary
132	TJ Olson	2100.00
128	Steven Markle	2200.00
136	Hazel Philtanker	2200.00
127	James Landry	2400.00
135	Ki Gee	2400.00
119	Karen Colmenares	2500.00
121	John Saito	2500.00
123	David Williams	2500.00
125	Michael Chang	2500.00
126	Patricia May	2500.00
129	John Williams	2500.00
130	Robert Davis	2500.00
131	John Williams	2500.00

Result 11 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	14:33:58	SELECT CONCAT(first_name, ' ', last_name) AS "names", salary, (0.15*salary) AS PF FROM e...	107 row(s) returned	0.000 sec / 0.000 sec

```
28 FROM employees;
29
30 /* 5) Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary */
31
32 • SELECT employee_id, CONCAT(first_name, ' ', last_name) AS "Names", salary
33 FROM employees
34 ORDER BY salary;
35
36 /* 6) Write a query to get the total salaries payable to employees */
37
38 • SELECT SUM(salary) AS "total salary"
39 FROM employees;
40
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

total salary
691400.00

Result 12 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	14:34:31	SELECT employee_id, CONCAT(first_name, ' ', last_name) AS "Names", salary FROM employees;	107 row(s) returned	0.000 sec / 0.000 sec

```
33     FROM employees
34     ORDER BY salary;
35
36 /* 6) Write a query to get the total salaries payable to employees */
37
38 • SELECT SUM(salary) AS "total salary"
39   FROM employees;
40
41 /* 7) Write a query to get the maximum and minimum salary from employees table */
42
43 • SELECT MAX(salary), MIN(salary)
44   FROM employees;
45
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

MAX(salary)	MIN(salary)
24000.00	2100.00

Result Grid

Form Editor

Result 13 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
7	14:35:38	SELECT SUM(salary) AS "total salary" FROM employees LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec

```
38 •   SELECT SUM(salary) AS "total salary"
39   FROM employees;
40
41 /* 7) Write a query to get the maximum and minimum salary from employees table */
42
43 •   SELECT MAX(salary), MIN(salary)
44   FROM employees;
45
46 /* 8) Write a query to get the average salary and number of employees in the employees table */
47
48 •   SELECT AVG(salary), COUNT(employee_id)
49   FROM employees;
50
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

AVG(salary)	COUNT(employee_id)
6461.682243	107

Result 14 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
8	14:36:23	SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

```
43 •   SELECT MAX(salary), MIN(salary)
44     FROM employees;
45
46     /* 8) Write a query to get the average salary and number of employees in the employees table */
47
48 •   SELECT AVG(salary), COUNT(employee_id)
49     FROM employees;
50
51     /* 9) Write a query to get the number of employees working with the company */
52
53 •   SELECT COUNT(employee_id)
54     FROM employees;
55
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

COUNT(employee_id)
107

Result 15 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	14:37:02	SELECT AVG(salary), COUNT(employee_id) FROM employees LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

```
48 •   SELECT AVG(salary), COUNT(employee_id)
49     FROM employees;
50
51    /* 9) Write a query to get the number of employees working with the company */
52
53 •   SELECT COUNT(employee_id)
54     FROM employees;
55
56    /* 10) Write a query to get the number of jobs available in the employees table */
57
58 •   SELECT COUNT(DISTINCT(job_id))
59     FROM employees;
60
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

COUNT(DISTINCT(job_id))
19

Result 16 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	14:37:37	SELECT COUNT(employee_id) FROM employees LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec

```
53 •  SELECT COUNT(employee_id)
54   FROM employees;
55
56   /* 10) Write a query to get the number of jobs available in the employees table */
57
58 •  SELECT COUNT(DISTINCT(job_id))
59   FROM employees;
60
61   /* 11) Write a query get all first name from employees table in upper case */
62
63 •  SELECT UPPER(first_name)
64   FROM employees;
65
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

UPPER(first_name)
STEVEN
NEENA
LEX
ALEXANDER
BRUCE
DAVID
.....

Result 17 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
11	14:38:04	SELECT COUNT(DISTINCT(job_id)) FROM employees LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

```
59     FROM employees;  
60  
61     /* 11) Write a query get all first name from employees table in upper case */  
62  
63 •  SELECT UPPER(first_name)  
64     FROM employees;  
65  
66     /* 12) Write a query to get the first 3 characters of first name from employees table */  
67  
68 •  SELECT first_name,  
69     LEFT(first_name, 3)  
70     FROM employees;  
71
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	first_name	LEFT(first_name, 3)
▶	Steven	Ste
	Neena	Nee
	Lex	Lex
	Alexander	Ale
	Bruce	Bru
	David	Dav

Result 18 Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	12 14:38:41	SELECT UPPER(first_name) FROM employees LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
64     FROM employees;  
65  
66     /* 12) Write a query to get the first 3 characters of first name from employees table */  
67  
68 •  SELECT first_name,  
69     LEFT(first_name, 3)  
70     FROM employees;  
71  
72     /* 13) Write a query to get first name from employees table after removing white spaces from both side */  
73  
74 •  SELECT TRIM(first_name)  
75     FROM employees;  
76
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

TRIM(first_name)
Steven
Neena
Lex
Alexander
Bruce
David
...

Result 19 Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
13	14:39:09	SELECT first_name, LEFT(first_name, 3) FROM employees LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
69    LEFT(first_name, 3)
70  FROM employees;
71
72  /* 13) Write a query to get first name from employees table after removing white spaces from both side */
73
74 •  SELECT TRIM(first_name)
75  FROM employees;
76
77  /* 14) Write a query to get the length of the employee names (first_name, last_name) from employees table */
78
79 •  SELECT first_name, last_name, CONCAT(first_name, ' ',last_name) , LENGTH(CONCAT(first_name,last_name))
80  FROM employees; |
81
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	first_name	last_name	CONCAT(first_name,' ',last_name)	LENGTH(CONCAT(first_name,last_name))
▶	Steven	King	Steven King	10
	Neena	Kochhar	Neena Kochhar	12
	Lex	De Haan	Lex De Haan	10
	Alexander	Hunold	Alexander Hunold	15
	Bruce	Ernst	Bruce Ernst	10
	David	Austin	David Austin	11

Result 20 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
14	14:39:44	SELECT TRIM(first_name) FROM employees LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
75      FROM employees;  
76  
77      /* 14) Write a query to get the length of the employee names (first_name, last_name) from employees table */  
78  
79 •  SELECT first_name, last_name, CONCAT(first_name, ' ', last_name), LENGTH(CONCAT(first_name, last_name))  
80      FROM employees;  
81  
82      /* 15) Write a query to check if the first_name fields of the employees table contains numbers */  
83  
84 •  SELECT first_name  
85      FROM employees  
86      WHERE first_name LIKE '(0-9)';  
87
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

first_name

employees 21 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
15	14:40:18	SELECT first_name, last_name, CONCAT(first_name, ' ', last_name), LENGTH(CONCAT(first_name, last_name)) FROM employees;	107 row(s) returned	0.015 sec / 0.000 sec

```
81
82     /* 15) Write a query to check if the first_name fields of the employees table contains numbers */
83
84 •   SELECT first_name
85     FROM employees
86     WHERE first_name LIKE '(0-9)';
87
88     /* 16) Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 */
89
90 •   SELECT CONCAT(first_name, ' ',last_name) AS "name", salary
91     FROM employees
92     WHERE salary<10000 OR salary>15000;
```

Result Grid				Filter Rows:
	name	salary		
▶	Steven King	24000.00		
	Neena Kochhar	17000.00		
	Lex De Haan	17000.00		
	Alexander Hunold	9000.00		
	Bruce Ernst	6000.00		
	David Austin	4800.00		
	Mark Chen	4200.00		

Result 22 ×

! Read Only

## Output

## Action Output

#	Time	Action	Message	Duration / Fetch
16	14:40:51	SELECT first_name FROM employees WHERE first_name LIKE '(0-9)' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

```
88     /* 16) Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 */
89
90 •  SELECT CONCAT(first_name, ' ', last_name) AS "name", salary
91   FROM employees
92   WHERE salary < 10000 OR salary > 15000;
93
94     /* 17) Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order */
95
96 •  SELECT first_name, last_name, department_id
97   FROM employees
98   WHERE department_id IN (30, 100)
99   ORDER BY department_id;
100
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

first_name	last_name	department_id
Den	Raphaely	30
Alexander	Khoo	30
Shelli	Baida	30
Sigal	Tobias	30
Guy	Himuro	30
Karen	Colmenares	30

employees 23 X

Result Grid

Form Editor

Action Output

#	Time	Action	Message	Duration / Fetch
17	14:41:22	SELECT CONCAT(first_name, ' ', last_name) AS "name", salary FROM employees WHERE sal...	91 row(s) returned	0.000 sec / 0.000 sec

```
94    /* 17) Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order */  
95  
96 • SELECT first_name, last_name, department_id  
97   FROM employees  
98   WHERE department_id IN (30, 100)  
99   ORDER BY department_id;  
100  
101  /* 18) Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 at  
102  
103 • SELECT first_name, last_name, salary, department_id  
104   FROM employees  
105   WHERE salary NOT BETWEEN 10000 AND 15000 AND department_id IN (30, 100);  
106
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	first_name	last_name	salary	department_id
▶	Alexander	Khoo	3100.00	30
	Shelli	Baida	2900.00	30
	Sigal	Tobias	2800.00	30
	Guy	Himuro	2600.00	30
	Karen	Colmenares	2500.00	30
	Daniel	Faviet	9000.00	100
			2800.00	100

employees 24 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	14:41:56	SELECT first_name, last_name, department_id FROM employees WHERE department_id IN (...)	12 row(s) returned	0.000 sec / 0.016 sec

```
100
101 /* 18) Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 and
102
103 • SELECT first_name, last_name, salary, department_id
104   FROM employees
105 WHERE salary NOT BETWEEN 10000 AND 15000 AND department_id IN (30, 100);
106
107 /* 19) Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987 */
108
109 • SELECT first_name, last_name, hire_date
110   FROM employees
111 WHERE YEAR (hire_date) = '1987';
112
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	first_name	last_name	hire_date
▶	Steven	King	1987-06-17
	Jennifer	Whalen	1987-09-17



employees 25 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
19	14:42:53	SELECT first_name, last_name, salary, department_id FROM employees WHERE salary NO...	10 row(s) returned	0.000 sec / 0.000 sec

```
105
106  /* 19) Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987 */
107
108 • SELECT first_name, last_name, hire_date
109   FROM employees
110  WHERE YEAR (hire_date) = '1987';
111
112  /* 20) Write a query to display the first_name of all employees who have both "b" and "c" in their first name */
113
114 • SELECT first_name
115   FROM employees
116  WHERE first_name  LIKE '%b%' AND first_name  LIKE '%c%';
117
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

first_name
Bruce

employees 20 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
20	14:48:45	SELECT first_name, last_name, hire_date FROM employees WHERE YEAR (hire_date) = '19... 2 row(s) returned		0.000 sec / 0.000 sec

```
111
112     /* 20) Write a query to display the first_name of all employees who have both "b" and "c" in their first name */
113
114 • SELECT first_name
115   FROM employees
116  WHERE first_name  LIKE '%b%' AND first_name  LIKE '%c%';
117
118     /* 21) Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary
119
120 • SELECT last_name, job_id, salary
121   FROM employees
122  WHERE job_id IN ('it_prof', 'sh_clerk') AND salary NOT IN (4500, 10000, 15000);
123
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	last_name	job_id	salary
▶	Taylor	SH_CLERK	3200.00
	Fleur	SH_CLERK	3100.00
	Sullivan	SH_CLERK	2500.00
	Geoni	SH_CLERK	2800.00
	Sarchand	SH_CLERK	4200.00
	Bull	SH_CLERK	4100.00
	" "	ANALYST	11000.00

employees 21 X

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
21	14:49:03	SELECT first_name FROM employees WHERE first_name  LIKE '%b%' AND first_name  LIKE ...	1 row(s) returned	0.000 sec / 0.000 sec

```
117
118     /* 21) Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary
119
120 •   SELECT last_name, job_id, salary
121     FROM employees
122     WHERE job_id IN ('it_prof', 'sh_clerk') AND salary NOT IN (4500, 10000, 15000);
123
124     /* 22) Write a query to display the last name of employees whose names have exactly 6 characters */
125
126 •   SELECT last_name
127     FROM employees
128     WHERE LENGTH(last_name) = 6;
129
```

< Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

last_name
Hunold
Austin
Faviet
Tobias
Himuro
Landry
...
employees 22 ×

Result Grid

Form Editor

Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
22	14:50:05	SELECT last_name, job_id, salary FROM employees WHERE job_id IN ('it_prof', 'sh_clerk') A...	20 row(s) returned	0.000 sec / 0.000 sec

```
123
124      /* 22) Write a query to display the last name of employees whose names have exactly 6 characters */
125
126 •  SELECT last_name
127     FROM employees
128    WHERE LENGTH(last_name) = 6;
129
130      /* 23) Write a query to display the last name of employees having 'e' as the third character */
131
132 •  SELECT last_name
133     FROM employees
134    WHERE last_name LIKE '_e%';
135
```

< Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

last_name
Greenberg
Chen
Gee
McEwen
Greene
Lee

employees 23 ×

Result Grid

Form Editor

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
23	14:50:43	SELECT last_name FROM employees WHERE LENGTH(last_name) = 6 LIMIT 0, 1000	28 row(s) returned	0.000 sec / 0.000 sec

```
135
136 /* 24) Write a query to get the job_id and related employee's id
137 Partial output of the query :
138   job_id      Employees ID
139   AC_ACCOUNT    206
140   AC_MGR        205
141   AD_ASST       200
142   AD_PRES       100
143   AD_VP         101 ,102
144   FI_ACCOUNT    110 ,113 ,111 ,109 ,112 */
145
146 • SELECT job_id, GROUP_CONCAT(employee_id SEPARATOR ',')
147   FROM employees GROUP BY job_id;
148
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

job_id	GROUP_CONCAT(employee_id SEPARATOR ',')
AC_ACCOUNT	206
AC_MGR	205
AD_ASST	200
AD_PRES	100

Result 24 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
24	14:51:17	SELECT last_name FROM employees WHERE last_name LIKE '__e%' LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec

```
145
146 •  SELECT job_id, GROUP_CONCAT(employee_id SEPARATOR ',')
147   FROM employees GROUP BY job_id;
148
149 /* 25) Write a query to update the portion of the phone_number in the employees table, within the phone number the substring '124' will be replaced by '?'
150
151 • UPDATE employees
152   SET phone_number = REPLACE(phone_number, '124', '999');
153
154 • SELECT *
155   FROM employees
156   WHERE phone_number LIKE '%999%';
157
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
▶	108	Nancy	Greenberg	NGREENBE@example.com	515.999.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
	109	Daniel	Faviet	DFAVIET@example.com	515.999.4169	1994-08-16	FI_ACCOUNT	9000.00	NULL	108	100
	110	John	Chen	JCHEN@example.com	515.999.4269	1997-09-28	FI_ACCOUNT	8200.00	NULL	108	100
	111	Ismael	Sciarra	ISCIARRA@example.com	515.999.4369	1997-09-30	FI_ACCOUNT	7700.00	NULL	108	100
	112	Jose Manuel	Urman	JMURMAN@example.com	515.999.4469	1998-03-07	FI_ACCOUNT	7800.00	NULL	108	100

employees 25 × Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
27	14:54:17	UPDATE employees SET phone_number = REPLACE(phone_number, '124', '999')	0 row(s) affected Rows matched: 107 Changed: 0 Warnings: 0	0.000 sec
28	14:54:17	SELECT * FROM employees WHERE phone_number LIKE '%999%' LIMIT 0, 1000	14 row(s) returned	0.000 sec / 0.000 sec

```

150
151 • UPDATE employees
152     SET phone_number = REPLACE(phone_number, '124', '999');
153
154 • SELECT *
155     FROM employees
156     WHERE phone_number LIKE '%999%';
157
158 /* 26) Write a query to get the details of the employees where the length of the first name greater than or equal to 8 */
159
160 • SELECT *
161     FROM employees
162     WHERE LENGTH(first_name) >= 8;
163

```

**Result Grid** | Filter Rows:  | Edit: Export/Import: | Wrap Cell Content:

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
▶	103	Alexander	Hunold	AHUNOLD@example.com	590.423.4567	1990-01-03	IT_PROG	9000.00	NULL	102	60
	112	Jose Manuel	Urman	JMURMAN@example.com	515.999.4469	1998-03-07	FI_ACCOUNT	7800.00	NULL	108	100
	115	Alexander	Khoo	AKHOO@example.com	515.127.4562	1995-05-18	PU_CLERK	3100.00	NULL	114	30
	153	Christopher	Olsen	COLSEN@example.com	011.44.1344.498718	1998-03-30	SA_REP	8000.00	0.20	145	80
	163	Danielle	Greene	DGREENE@example.com	011.44.1346.229268	1999-03-19	SA_REP	9500.00	0.15	147	80
	164	Harrison	Ridnm	HRI DNM@example.com	011 44 1343 829268	1998-03-23	SA_RFP	10000.00	0.20	148	80

employees 27 X

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
29	14:55:57	SELECT * FROM employees WHERE LENGTH(first_name) >= 8 LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec

```
159
160 •   SELECT *
161     FROM employees
162     WHERE LENGTH(first_name) >= 8;
163
164     /* 27) Write a query to append '@example.com' to email field */
165
166 •   SELECT email
167     FROM employees;
168
169 •   UPDATE employees
170     SET email = CONCAT(email, '@example.com');
171
```

< | Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ] |

email
▶ Steven@example.com
▶ Neena@example.com
▶ Lex@example.com
▶ Alexander@example.com
▶ Bruce@example.com

employees 31 X | Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
38	15:02:06	SELECT email FROM employees LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
164 /* 27) Write a query to append '@example.com' to email field */
165
166 • SELECT email
167   FROM employees;
168
169 • UPDATE employees
170   SET email = CONCAT(email, '@example.com');
171
172 /* 28) Write a query to extract the last 4 character of phone numbers */
173
174 • SELECT RIGHT(phone_number, 4) AS 'Phone No.'
175   FROM employees;
176
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

Phone No.
4567
4568
4569
4567
4568

Result 33 x

Output ::::::::::::::::::::

Action Output

#	Time	Action	Message	Duration / Fetch
40	15:02:53	SELECT RIGHT(phone_number, 4) AS 'Phone No.' FROM employees LIMIT 0, 1000	107 row(s) returned	0.062 sec / 0.000 sec

```
169 • UPDATE employees
170     SET email = CONCAT(email, '@example.com');
171
172     /* 28) Write a query to extract the last 4 character of phone numbers */
173
174 • SELECT RIGHT(phone_number, 4) AS 'Phone No.'
175     FROM employees;
176
177     /* 29) Write a query to get the last word of the street address */
178
179 • SELECT location_id, street_address, substring_index(street_address, " ", -1) AS 'Last-word-of-street_address'
180     FROM locations;
181
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	location_id	street_address	Last-word-of-street_address
▶	1000	1297 Via Cola di Rie	Rie
	1100	93091 Calle della Testa	Testa
	1200	2017 Shinjuku-ku	Shinjuku-ku
	1300	9450 Kamiya-cho	Kamiya-cho
	1400	2014 Jabberwocky Rd	Rd

Result 34 × Read Only

Output ::::::::::::

Action Output

#	Time	Action	Message	Duration / Fetch
41	15:03:03	SELECT RIGHT(phone_number, 4) AS 'Phone No.' FROM employees LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
FROM employees;

/* 29) Write a query to get the last word of the street address */

• 179 SELECT location_id, street_address, substring_index(street_address, " ", -1) AS 'Last--word-of-street_address'
180 FROM locations;

/* 30) Write a query to get the locations that have minimum street length */

• 184 SELECT * FROM locations
185 WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(street_address))
186 FROM locations);
187
```

Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

	location_id	street_address	postal_code	city	state_province	country_id
▶	1600	2007 Zagora St	50090	South Brunswick	New Jersey	US
	2400	8204 Arthur St	NULL	London	NULL	UK
*	NULL	NULL	NULL	NULL	NULL	NULL

locations 35 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
42	15:03:48	SELECT location_id, street_address, substring_index(street_address, " ", -1) AS 'Last--word-of... 23 row(s) returned		0.016 sec / 0.000 sec

```
180    FROM locations;
181
182    /* 30) Write a query to get the locations that have minimum street length */
183
184 •   SELECT * FROM locations
185     WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(street_address))
186     FROM locations);
187
188    /* 31) Write a query to display the first word from those job titles which contains more than one words */
189
190 •   SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, ' '))
191     FROM jobs;
192
```

Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

job_title	SUBSTR(job_title,1, INSTR(job_title, ' '))
Public Accountant	Public
Accounting Manager	Accounting
Administration Assistant	Administration
President	

Result 37 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
44	15:05:35	SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, ' ')) FROM jobs LIMIT 0, 1000	19 row(s) returned	0.062 sec / 0.000 sec

```
186     FROM locations);  
187  
188     /* 31) Write a query to display the first word from those job titles which contains more than one words */  
189  
190 •  SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, ' '))  
191     FROM jobs;  
192  
193     /* 32) Write a query to display the length of first name for employees where last name contain character 'c' after 2nd position */  
194  
195 •  SELECT first_name, last_name, LENGTH(first_name)  
196     FROM employees  
197     WHERE instr(last_name,'C') > 2;  
198  
  
Result Grid | Filter Rows:  | Export:  | Wrap Cell Content:   


|   | first_name | last_name | LENGTH(first_name) |
|---|------------|-----------|--------------------|
| ▶ | Neena      | Kochhar   | 5                  |
|   | Peter      | Tucker    | 5                  |
|   | Nandita    | Sarchand  | 7                  |

  
Result 39 x Read Only  
Output :  
Action Output  


| #  | Time     | Action                                                                                | Message           | Duration / Fetch      |
|----|----------|---------------------------------------------------------------------------------------|-------------------|-----------------------|
| 46 | 15:06:11 | SELECT first_name, last_name, LENGTH(first_name) FROM employees WHERE instr(last_n... | 3 row(s) returned | 0.000 sec / 0.000 sec |


```

```
192
193     /* 32) Write a query to display the length of first name for employees where last name contain character 'c' after 2nd position */
194
195 •  SELECT first_name, last_name, LENGTH(first_name)
196   FROM employees
197  WHERE instr(last_name,'C') > 2;
198
199     /* 33) Write a query that displays the first name and the length of the first name for all employees whose name starts with the letters 'A', 'J' or 'M'.
200
201 •  SELECT first_name AS "Name", LENGTH(first_name)  AS "Length"
202   FROM employees
203  WHERE first_name LIKE 'J%' OR first_name LIKE 'M%' OR first_name LIKE 'A%' ORDER BY first_name ;
204
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

Name	Length
Adam	4
Alana	5
Alberto	7
Alexander	9
Alexander	9

Result 41 x Read Only

Output ::::::::::::

Action Output

#	Time	Action	Message	Duration / Fetch
48	15:07:22	SELECT first_name AS "Name", LENGTH(first_name) AS "Length" FROM employees WHE...	32 row(s) returned	0.000 sec / 0.000 sec

```
197 WHERE instr(last_name,'C') > 2;
198
199 /* 33) Write a query that displays the first name and the length of the first name for all employees whose name starts with the letters 'A', 'J' or 'M'.
200
201 • SELECT first_name AS "Name", LENGTH(first_name) AS "Length"
202   FROM employees
203   WHERE first_name LIKE 'J%' OR first_name LIKE 'M%' OR first_name LIKE 'A%' ORDER BY first_name ;
204
205 /* 34) Write a query to display the first name and salary for all employees. Format the salary to be 10 characters long, left-padded with the $ symbol. I
206
207 • SELECT first_name, LPAD(salary, 10, '$') AS salary
208   FROM employees;
209
```

< >

Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

	first_name	salary
▶	Steven	\$24000.00
	Neena	\$17000.00
	Lex	\$17000.00
	Alexander	\$\$9000.00
	Bruce	\$\$\$6000.00

Result 43 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	15:08:01	SELECT first_name, LPAD(salary, 10, '\$') AS salary FROM employees LIMIT 0, 1000	107 row(s) returned	0.000 sec / 0.000 sec

```
FROM employees
WHERE first_name LIKE 'J%' OR first_name LIKE 'M%' OR first_name LIKE 'A%' ORDER BY first_name ;
/* 34) Write a query to display the first name and salary for all employees. Format the salary to be 10 characters long, left-padded with the $ symbol. */
SELECT first_name, LPAD(salary, 10, '$') AS salary
FROM employees;
/* 35) Write a query to display the first eight characters of the employees' first names and indicates the amounts of their salaries with '$' sign. Each employee's salary is to be rounded off to the nearest integer value. */
SELECT LEFT(first_name, 8) AS first_name, REPEAT('$', FLOOR(salary/1000)) AS 'SALARY($)', salary
FROM employees ORDER BY salary DESC;
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	first_name	SALARY(\$)	salary
▶	Steven	\$\$\$\$\$\$\$\$\$\$\$\$\$\$	24000.00
	Neena	\$\$\$\$\$\$\$\$\$\$	17000.00
	Lex	\$\$\$\$\$\$\$\$\$\$	17000.00
	John	\$\$\$\$\$\$\$\$\$\$	14000.00
	Karen	\$\$\$\$\$\$\$\$\$\$	13500.00

Result 45 ×      Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
52	15:08:47	SELECT LEFT(first_name, 8) AS first_name, REPEAT('\$', FLOOR(salary/1000)) AS 'SALARY(...'	107 row(s) returned	0.000 sec / 0.000 sec

```
209
210     /* 35) Write a query to display the first eight characters of the employees' first names and indicates the amounts of their salaries with '$' sign. Each
211
212 • SELECT LEFT(first_name, 8) AS first_name, REPEAT('$', FLOOR(salary/1000)) AS 'SALARY($)', salary
213     FROM employees ORDER BY salary DESC;
214
215 • /* 36) Write a query to display the employees with their code, first name, last name and hire date who hired
216     either on seventh day of any month or seventh month in any year */
217
218 • SELECT employee_id, first_name, last_name, hire_date
219     FROM employees
220     WHERE DAY(hire_date) = 7 OR MONTH(hire_date) = 7;
```

Result Grid | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	employee_id	first_name	last_name	hire_date
▶	107	Diana	Lorentz	1999-02-07
	112	Jose Manuel	Urman	1998-03-07
	113	Luis	Popp	1999-12-07
	114	Den	Raphaely	1994-12-07
	117	Sigal	Tobias	1997-07-24

employees 47 ×

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
54	15:09:31	SELECT employee_id, first_name, last_name, hire_date FROM employees WHERE DAY(hire...	16 row(s) returned	0.000 sec / 0.000 sec

```
218 • SELECT employee_id, first_name, last_name, hire_date
219   FROM employees
220 WHERE DAY(hire_date) = 7 OR MONTH(hire_date) = 7;
221
222
223 /*-----northwind DATABASE EXERCISE-----*/
224
225 • USE northwind;
226
227 /* 1) Write a query to get Product name and quantity/unit */
228
229 • SELECT ProductName, QuantityPerUnit
230   FROM Products;
231
232 /* 2) Write a query to get current Product list (Product ID and name) */
233
234 • SELECT ProductID, ProductName
235   FROM Products
236 WHERE Discontinued = "0"
237 ORDER BY ProductName;
```

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
56	15:10:07	USE northwind	0 row(s) affected	0.000 sec

```
219 FROM employees  
220 WHERE DAY(hire_date) = 7 OR MONTH(hire_date) = 7;  
221  
222  
223 /*-----northwind DATABASE EXERCISE-----*/  
224  
225 • USE northwind;  
226  
227 /* 1) Write a query to get Product name and quantity/unit */  
228  
229 • SELECT ProductName, QuantityPerUnit  
230 FROM Products;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

ProductName	QuantityPerUnit
Chai	10 boxes x 20 bags
Chang	24 - 12 oz bottles
Aniseed Syrup	12 - 550 ml bottles
Chef Anton's Cajun Seasoning	48 - 6 oz jars
Chef Anton's Gumbo Mix	36 boxes

Products 49 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
58	15:12:19	SELECT ProductName, QuantityPerUnit FROM Products LIMIT 0, 1000	77 row(s) returned	0.000 sec / 0.000 sec

```
225 • USE northwind;
226
227 /* 1) Write a query to get Product name and quantity/unit */
228
229 • SELECT ProductName, QuantityPerUnit
230   FROM Products;
231
232 /* 2) Write a query to get current Product list (Product ID and name) */
233
234 • SELECT ProductID, ProductName
235   FROM Products
236   WHERE Discontinued = "0" ORDER BY ProductName;
237
```

Result Grid | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	ProductID	ProductName
▶	3	Aniseed Syrup
▶	40	Boston Crab Meat
▶	60	Camembert Pierrot
▶	18	Carnarvon Tigers
▶	1	Chai

Products 57 x

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
66	15:15:25	SELECT ProductID, ProductName FROM Products WHERE Discontinued = "0" ORDER BY ...	69 row(s) returned	0.000 sec / 0.000 sec

```
231
232 /* 2) Write a query to get current Product list (Product ID and name) */
233
234 • SELECT ProductID, ProductName
235   FROM Products
236 WHERE Discontinued = "0" ORDER BY ProductName;
237
238 /* 3) Write a query to get discontinued Product list (Product ID and name) */
239
240 • SELECT ProductID, ProductName
241   FROM Products
242 WHERE Discontinued = "1" ORDER BY ProductName;
243
```

Result Grid | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	ProductID	ProductName
▶	17	Alice Mutton
	5	Chef Anton's Gumbo Mix
▶	24	Guaraná Fantástica
	9	Mishi Kobe Niku
	53	Perth Pasties

Products 59 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
68	15:16:13	SELECT ProductID, ProductName FROM Products WHERE Discontinued = "1" ORDER BY ...	8 row(s) returned	0.000 sec / 0.000 sec

```
236 WHERE Discontinued = "0" ORDER BY ProductName;
237
238 /* 3) Write a query to get discontinued Product list (Product ID and name) */
239
240 • SELECT ProductID, ProductName
241 FROM Products
242 WHERE Discontinued = "1" ORDER BY ProductName;
243
244 /* 4) Write a query to get most expense and least expensive Product list (name and unit price) */
245
246 • SELECT ProductName, UnitPrice
247 FROM Products ORDER BY UnitPrice DESC;
248
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	ProductName	UnitPrice
▶	Côte de Blaye	263.5000
	Thüringer Rostbratwurst	123.7900
	Mishi Kobe Niku	97.0000
	Sir Rodney's Marmalade	81.0000
	Carnarvon Tigers	62.5000

Products 61 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
70	15:16:52	SELECT ProductName, UnitPrice FROM Products ORDER BY UnitPrice DESC LIMIT 0, 1000	77 row(s) returned	0.000 sec / 0.000 sec

```
242 WHERE Discontinued = "1" ORDER BY ProductName;
243
244 /* 4) Write a query to get most expensive and least expensive Product list (name and unit price) */
245
246 • SELECT ProductName, UnitPrice
247 FROM Products ORDER BY UnitPrice DESC;
248
249 /* 5) Write a query to get Product list (id, name, unit price) where current products cost less than $20 */
250
251 • SELECT ProductID, ProductName, UnitPrice
252 FROM Products
253 WHERE (((UnitPrice) < 20) AND ((Discontinued) = 0)) ORDER BY UnitPrice DESC;
254
```

Result Grid | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	ProductID	ProductName	UnitPrice
▶	57	Ravioli Angelo	19.5000
	44	Gula Malacca	19.4500
	2	Chang	19.0000
	36	Inlagd Sill	19.0000
	40	Boston Crab Meat	18.4000

Products 63 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
72	15:17:27	SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (((UnitPrice) < 20) AN...	37 row(s) returned	0.000 sec / 0.000 sec

```
248
249     /* 5) Write a query to get Product list (id, name, unit price) where current products cost less than $20 */
250
251 • SELECT ProductID, ProductName, UnitPrice
252     FROM Products
253     WHERE ((UnitPrice) < 20) AND ((Discontinued) = 0)) ORDER BY UnitPrice DESC;
254
255     /* 6) Write a query to get Product list (id, name, unit price) where products cost between $15 and $25 */
256
257 • SELECT ProductID, ProductName, UnitPrice
258     FROM Products
259     WHERE ((UnitPrice) >= 15 AND (UnitPrice) <= 25);
260
```

Result Grid | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	ProductID	ProductName	UnitPrice
▶	1	Chai	18.0000
	2	Chang	19.0000
	4	Chef Anton's Cajun Seasoning	22.0000
	5	Chef Anton's Gumbo Mix	21.3500
	6	Grandma's Boysenberry Spread	25.0000

Products 65 x

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
74	15:17:58	SELECT ProductID, ProductName, UnitPrice FROM Products WHERE ((UnitPrice) >= 15 AN...	25 row(s) returned	0.000 sec / 0.000 sec

```
254  
255 /* 6) Write a query to get Product list (id, name, unit price) where products cost between $15 and $25 */  
256  
257 • SELECT ProductID, ProductName, UnitPrice  
FROM Products  
WHERE ((UnitPrice) >= 15 AND (UnitPrice) <= 25);  
260  
261 /* 7) Write a query to get Product list (name, unit price) of above average price */  
262  
263 • SELECT ProductName, UnitPrice  
FROM Products  
WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM products);  
266
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	ProductName	UnitPrice
▶	Uncle Bob's Organic Dried Pears	30.0000
	Northwoods Cranberry Sauce	40.0000
	Mishi Kobe Niku	97.0000
	Ikura	31.0000
	Queso Manchego La Pastora	38.0000

Products 67 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
76	15:18:41	SELECT ProductName, UnitPrice FROM Products WHERE UnitPrice > (SELECT AVG(UnitPr...)	25 row(s) returned	0.016 sec / 0.000 sec

```
259 WHERE ((UnitPrice) >= 15 AND (UnitPrice) <= 25);
260
261 /* 7) Write a query to get Product list (name, unit price) of above average price */
262
263 • SELECT ProductName, UnitPrice
264   FROM Products
265 WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM products);
266
267 /* 8) Write a query to get Product list (name, unit price) of ten most expensive products */
268
269 • SELECT ProductID, ProductName, UnitPrice
270   FROM products ORDER BY UnitPrice DESC LIMIT 10;
271
```

Result Grid				Filter Rows:
	ProductID	ProductName	UnitPrice	
▶	38	Côte de Blaye	263.5000	
	29	Thüringer Rostbratwurst	123.7900	
	9	Mishi Kobe Niku	97.0000	
	20	Sir Rodney's Marmalade	81.0000	
	18	Carnarvon Tigers	62.5000	

products 69 ×

#### Output :

Action Output				
#	Time	Action	Message	Duration / Fetch
78	15:19:24	SELECT ProductID, ProductName, UnitPrice FROM products ORDER BY UnitPrice DESC LI...	10 row(s) returned	0.016 sec / 0.000 sec

```
/* 8) Write a query to get Product list (name, unit price) of ten most expensive products */
SELECT ProductID, ProductName, UnitPrice
FROM products ORDER BY UnitPrice DESC LIMIT 10;

/* 9) Write a query to count current and discontinued products */

SELECT(CASE WHEN Discontinued = 0 THEN 'Current_Products_Count'
            ELSE 'Discontinued_Products_Count'
        END) AS 'current & discontinued products',
       COUNT(ProductName) AS 'Products_Count'
FROM Products GROUP BY Discontinued;
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	current & discontinued products	Products_Count
▶	Current_Products_Count	69
	Discontinued_Products_Count	8

Result 71 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
80	15:24:56	SELECT(CASE WHEN Discontinued = 0 THEN 'Current_Products_Count' ELSE 'Disc...' 2 row(s) returned		0.016 sec / 0.000 sec

```
273
274 • SELECT(CASE WHEN Discontinued = 0 THEN 'Current_Products_Count'
275     ELSE 'Discontinued_Products_Count'
276     END) AS 'current & discontinued products',
277     COUNT(ProductName) AS 'Products_Count'
278 FROM Products GROUP BY Discontinued;
279
280 /* 10) Write a query to get Product list (name, units on order , units in stock) of stock is less than the quantity on order */
281
282 • SELECT ProductName, UnitsOnOrder, UnitsInStock
283 FROM Products
284 WHERE ((UnitsInStock) < UnitsOnOrder);
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

ProductName	UnitsOnOrder	UnitsInStock
Chang	40	17
Aniseed Syrup	70	13
Queso Cabrales	30	22
Sir Rodney's Scones	40	3
Gorgonzola Telino	70	0

Products 73 ×

Output ::::::::::::

Action Output

#	Time	Action	Message	Duration / Fetch
82	15:25:36	SELECT ProductName, UnitsOnOrder, UnitsInStock FROM Products WHERE ((UnitsInStock...)	14 row(s) returned	0.000 sec / 0.000 sec