

Document highlight:

- JOIN in MySQL
- UNION in MySQL
- String Functions in MySQL
 - ie., Length, Upper, Lower, Trim, Left, Right, Substring, Replace, Locate, Concat
- CASE Statements in MySQL
- Subqueries in MySQL
- Window Functions in MySQL
 - ie., Clause - Over, Partition By, Order By
 - ie., Rank, Dense Rank, Row Number

1. Setup MySQL Workbench (community)

Download: and install (follow installation guide):

<https://dev.mysql.com/downloads/installer/>

The screenshot shows the MySQL Community Downloads page for the MySQL Installer. The page has a dark header with the MySQL logo and the text "MySQL Community Downloads". Below the header, there is a breadcrumb trail: "MySQL Installer". The main content area has a tabbed interface with "General Availability (GA) Releases" selected. The title "MySQL Installer 8.0.40" is displayed. A note states: "Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server." Below the note, there are two dropdown menus: "Select Version:" with "8.0.40" selected, and "Select Operating System:" with "Microsoft Windows" selected. A table lists the available installers:

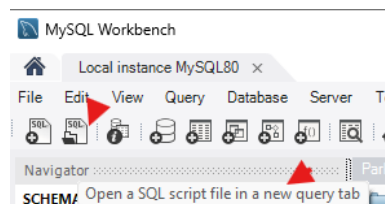
Operating System	Version	Size	Action
Windows (x86, 32-bit), MSI Installer	8.0.40	2.1 M	Download
<small>(mysql-installer-web-community-8.0.40.0.msi) MD5: 42da0dc06ad328fe2451eeb3998fb016 Signature</small>			
Windows (x86, 32-bit), MSI Installer	8.0.40	306.4M	Download
<small>(mysql-installer-community-8.0.40.0.msi) MD5: 8c1bf3a285d5e191e36dc334a10f55d2 Signature</small>			

Below the table, a note says: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download."

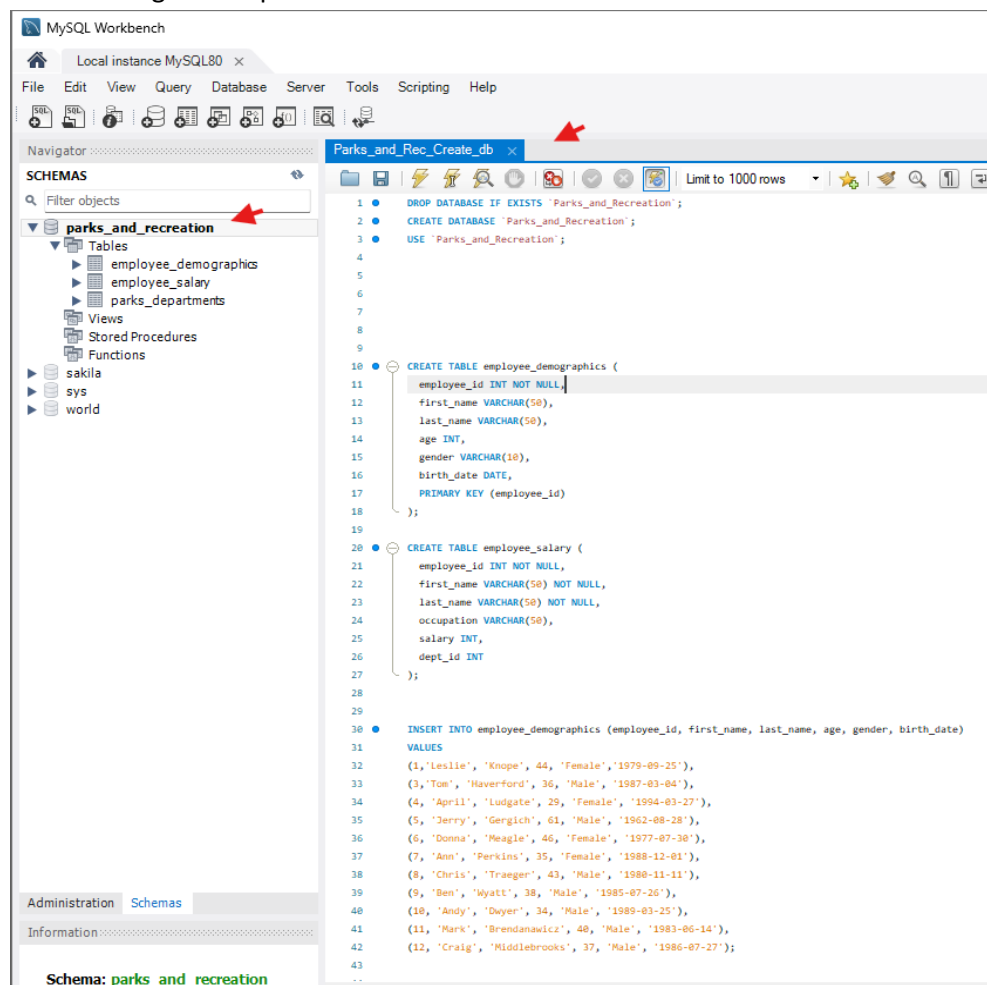
MySQL home screen



2. Create a database (create tables and load the set of data into it)



After running the script:



3. Main reference tables

SCHEMAS

Filter objects

- ▼ parks_and_recreation
 - ▼ Tables
 - employee_demographics**
 - employee_salary
 - parks_departments
 - Views
 - Stored Procedures
 - Functions
- ▼ sakila
- ▼ sys
- ▼ world

```
1 #This is selecting all value(s) to display from a table
2 # --Query 01
3 SELECT
4 *
5 FROM
6 parks_and_recreation.employee_demographics
7 ;
8
```

Result Grid | Filter Rows: | Edit: | Export/Import:

employee_id	first_name	last_name	age	gender	birth_date
1	Leslie	Knope	44	Female	1979-09-25
3	Tom	Haverford	36	Male	1987-03-04
4	April	Ludgate	29	Female	1994-03-27
5	Jerry	Gergich	61	Male	1962-08-28
6	Donna	Meagle	46	Female	1977-07-30
7	Ann	Perkins	35	Female	1988-12-01
8	Chris	Traeger	43	Male	1980-11-11
9	Ben	Wyatt	38	Male	1985-07-26
10	Andy	Dwyer	34	Male	1989-03-25
11	Mark	Brendanawicz	40	Male	1983-06-14
12	Craig	Middlebrooks	37	Male	1986-07-27
*	NULL	NULL	NULL	NULL	NULL

SCHEMAS

Filter objects

- ▼ parks_and_recreation
 - ▼ Tables
 - employee_demographics
 - employee_salary**
 - parks_departments
 - Views
 - Stored Procedures
 - Functions
- ▼ sakila
- ▼ sys
- ▼ world

```
8
9 #This is selecting all value(s) to display from a table
10 # --Query 01
11 SELECT
12 *
13 FROM
14 parks_and_recreation.employee_salary
15 ;
```

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

employee_id	first_name	last_name	occupation	salary	dept_id
1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1
2	Ron	Swanson	Director of Parks and Recreation	70000	1
3	Tom	Haverford	Entrepreneur	50000	1
4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1
5	Jerry	Gergich	Office Manager	50000	1
6	Donna	Meagle	Office Manager	60000	1
7	Ann	Perkins	Nurse	55000	4
8	Chris	Traeger	City Manager	90000	3
9	Ben	Wyatt	State Auditor	70000	6
10	Andy	Dwyer	Shoe Shiner and Musician	20000	NULL
11	Mark	Brendana...	City Planner	57000	3
12	Craig	Middlebro...	Parks Director	65000	1

SCHEMAS

Filter objects

- ▼ parks_and_recreation
 - ▼ Tables
 - employee_demographics
 - employee_salary
 - parks_departments**
 - Views
 - Stored Procedures
 - Functions
- ▼ sakila
- ▼ sys
- ▼ world

```
1 #This is selecting all value(s) to display from a table
2 # --Query 01
3 SELECT
4 *
5 FROM
6 parks_and_recreation.parks_departments
7 ;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

department_id	department_name
1	Parks and Recreation
2	Animal Control
3	Public Works
4	Healthcare
5	Library
6	Finance
*	NULL

4. JOIN in MySQL (flavors of queries and its sampling output)

01-JOIN in MySQL

```
1  #This is selecting value(s) to display from a table, using:
2  #      (a) INNER JOIN in MySQL
3  #      (b) OUTER JOIN in MySQL
4  #      (c) SELF JOIN in MySQL
5  #      (d) JOIN multiple tables in MySQL
```

```
7  # --Query 01 (sampling of inner join in MySQL)
8  • SELECT
9      *
10 FROM
11     parks_and_recreation.employee_demographics
12 INNER JOIN
13     parks_and_recreation.employee_salary
14 ON
15     employee_demographics.employee_id = employee_salary.employee_id
16 ;
```

Result Grid											
Filter Rows:											
Exports: Wrap Cell Content:											
employee_id	first_name	last_name	age	gender	birth_date	employee_id	first_name	last_name	occupation	salary	dept_id
1	Leslie	Knope	44	Female	1979-09-25	1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1
3	Tom	Haverford	36	Male	1987-03-04	3	Tom	Haverford	Entrepreneur	50000	1
4	April	Ludgate	29	Female	1994-03-27	4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1
5	Jerry	Gergich	61	Male	1962-08-28	5	Jerry	Gergich	Office Manager	50000	1
6	Donna	Meagle	46	Female	1977-07-30	6	Donna	Meagle	Office Manager	60000	1
7	Ann	Perkins	35	Female	1988-12-01	7	Ann	Perkins	Nurse	55000	4
8	Chris	Traeger	43	Male	1980-11-11	8	Chris	Traeger	City Manager	90000	3
9	Ben	Wyatt	38	Male	1985-07-26	9	Ben	Wyatt	State Auditor	70000	6
10	Andy	Dwyer	34	Male	1989-03-25	10	Andy	Dwyer	Shoe Shiner and Musician	20000	NULL
11	Mark	Brendanawicz	40	Male	1983-06-14	11	Mark	Brendanawicz	City Planner	57000	3
12	Craig	Middlebrooks	37	Male	1986-07-27	12	Craig	Middlebrooks	Parks Director	65000	1

```
18 # --Query 02 (sampling of inner join in MySQL)
19 • SELECT
20     *
21 FROM
22     parks_and_recreation.employee_demographics AS dem
23 INNER JOIN
24     parks_and_recreation.employee_salary AS sal
25 ON
26     dem.employee_id = sal.employee_id
27 ;
28
```

Result Grid											
Filter Rows:											
Exports: Wrap Cell Content:											
employee_id	first_name	last_name	age	gender	birth_date	employee_id	first_name	last_name	occupation	salary	dept_id
1	Leslie	Knope	44	Female	1979-09-25	1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1
3	Tom	Haverford	36	Male	1987-03-04	3	Tom	Haverford	Entrepreneur	50000	1
4	April	Ludgate	29	Female	1994-03-27	4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1
5	Jerry	Gergich	61	Male	1962-08-28	5	Jerry	Gergich	Office Manager	50000	1
6	Donna	Meagle	46	Female	1977-07-30	6	Donna	Meagle	Office Manager	60000	1
7	Ann	Perkins	35	Female	1988-12-01	7	Ann	Perkins	Nurse	55000	4
8	Chris	Traeger	43	Male	1980-11-11	8	Chris	Traeger	City Manager	90000	3
9	Ben	Wyatt	38	Male	1985-07-26	9	Ben	Wyatt	State Auditor	70000	6
10	Andy	Dwyer	34	Male	1989-03-25	10	Andy	Dwyer	Shoe Shiner and Musician	20000	NULL
11	Mark	Brendanawicz	40	Male	1983-06-14	11	Mark	Brendanawicz	City Planner	57000	3
12	Craig	Middlebrooks	37	Male	1986-07-27	12	Craig	Middlebrooks	Parks Director	65000	1

```
29 # --Query 03 (sampling of inner join in MySQL)
```

```
30 • SELECT
31     dem.employee_id AS ID,
32     age AS Age,
33     occupation AS Occupation,
34     sal.salary AS Salary
35 FROM
36     parks_and_recreation.employee_demographics AS dem
37 INNER JOIN
38     parks_and_recreation.employee_salary AS sal
39 ON
40     dem.employee_id = sal.employee_id
41 ;
```

Result Grid Filter Rows: Export: Wrap Cell Content: [fA](#)

	ID	Age	Occupation	Salary
▶	1	44	Deputy Director of Parks and Recreation	75000
	3	36	Entrepreneur	50000
	4	29	Assistant to the Director of Parks and Recreation	25000
	5	61	Office Manager	50000
	6	46	Office Manager	60000
	7	35	Nurse	55000
	8	43	City Manager	90000
	9	38	State Auditor	70000
	10	34	Shoe Shiner and Musician	20000
	11	40	City Planner	57000
	12	37	Parks Director	65000

```
43 # --Query 04 (sampling of outer join in MySQL)
```

```
44 • SELECT
45     *
46 FROM
47     parks_and_recreation.employee_demographics AS dem
48 LEFT OUTER JOIN
49     parks_and_recreation.employee_salary AS sal
50 ON
51     dem.employee_id = sal.employee_id
52 ;
```

Result Grid Filter Rows: Export: Wrap Cell Content: [fA](#)

	employee_id	first_name	last_name	age	gender	birth_date	employee_id	first_name	last_name	occupation	salary	dept_id
▶	1	Leslie	Knope	44	Female	1979-09-25	1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1
	3	Tom	Haverford	36	Male	1987-03-04	3	Tom	Haverford	Entrepreneur	50000	1
	4	April	Ludgate	29	Female	1994-03-27	4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1
	5	Jerry	Gergich	61	Male	1962-08-28	5	Jerry	Gergich	Office Manager	50000	1
	6	Donna	Meagle	46	Female	1977-07-30	6	Donna	Meagle	Office Manager	60000	1
	7	Ann	Perkins	35	Female	1988-12-01	7	Ann	Perkins	Nurse	55000	4
	8	Chris	Traeger	43	Male	1980-11-11	8	Chris	Traeger	City Manager	90000	3
	9	Ben	Wyatt	38	Male	1985-07-26	9	Ben	Wyatt	State Auditor	70000	6
	10	Andy	Dwyer	34	Male	1989-03-25	10	Andy	Dwyer	Shoe Shiner and Musician	20000	10000
	11	Mark	Brendanawicz	40	Male	1983-06-14	11	Mark	Brendanawicz	City Planner	57000	3
	12	Craig	Middlebrooks	37	Male	1986-07-27	12	Craig	Middlebrooks	Parks Director	65000	1

```

54 # --Query 05 (sampling of outer join in MySQL)
55 • SELECT
56 *
57 FROM
58 parks_and_recreation.employee_demographics AS dem
59 RIGHT OUTER JOIN
60 parks_and_recreation.employee_salary AS sal
61 ON
62 dem.employee_id = sal.employee_id
63

```

Result Grid											
Filter Rows:											
Export: Wrap Cell Content: I A											
employee_id	first_name	last_name	age	gender	birth_date	employee_id	first_name	last_name	occupation	salary	dept_id
1	Leslie	Knope	44	Female	1979-09-25	1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1
						2	Ron	Swanson	Director of Parks and Recreation	70000	1
3	Tom	Haverford	36	Male	1987-03-04	3	Tom	Haverford	Entrepreneur	50000	1
4	April	Ludgate	29	Female	1994-03-27	4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1
5	Jerry	Gergich	61	Male	1962-08-28	5	Jerry	Gergich	Office Manager	50000	1
6	Donna	Meagle	46	Female	1977-07-30	6	Donna	Meagle	Office Manager	60000	1
7	Ann	Perkins	35	Female	1988-12-01	7	Ann	Perkins	Nurse	55000	4
8	Chris	Traeger	43	Male	1980-11-11	8	Chris	Traeger	City Manager	90000	3
9	Ben	Wyatt	38	Male	1985-07-26	9	Ben	Wyatt	State Auditor	70000	6
10	Andy	Dwyer	34	Male	1989-03-25	10	Andy	Dwyer	Shoe Shiner and Musician	20000	
11	Mark	Brendana...	40	Male	1983-06-14	11	Mark	Brendanawicz	City Planner	57000	3
12	Craig	Middlebro...	37	Male	1986-07-27	12	Craig	Middlebrooks	Parks Director	65000	1

```

65 # --Query 06 (sampling of self join in MySQL)
66 • SELECT
67 *
68 FROM
69 parks_and_recreation.employee_salary AS emp1
70 JOIN
71 parks_and_recreation.employee_salary AS emp2
72 ON
73 emp1.employee_id + 1 = emp2.employee_id
74

```

Result Grid											
Filter Rows:											
Export: Wrap Cell Content: I A											
employee_id	first_name	last_name	occupation	salary	dept_id	employee_id	first_name	last_name	occupation	salary	dept_id
1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1	2	Ron	Swanson	Director of Parks and Recreation	70000	1
2	Ron	Swanson	Director of Parks and Recreation	70000	1	3	Tom	Haverford	Entrepreneur	50000	1
3	Tom	Haverford	Entrepreneur	50000	1	4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1
4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1	5	Jerry	Gergich	Office Manager	50000	1
5	Jerry	Gergich	Office Manager	50000	1	6	Donna	Meagle	Office Manager	60000	1
6	Donna	Meagle	Office Manager	60000	1	7	Ann	Perkins	Nurse	55000	4
7	Ann	Perkins	Nurse	55000	4	8	Chris	Traeger	City Manager	90000	3
8	Chris	Traeger	City Manager	90000	3	9	Ben	Wyatt	State Auditor	70000	6
9	Ben	Wyatt	State Auditor	70000	6	10	Andy	Dwyer	Shoe Shiner and Musician	20000	
10	Andy	Dwyer	Shoe Shiner and Musician	20000		11	Mark	Brendanawicz	City Planner	57000	3
11	Mark	Brendanawicz	City Planner	57000	3	12	Craig	Middlebrooks	Parks Director	65000	1

```
76 # --Query 07 (sampling of self join in MySQL)
```

```
77 • SELECT
78     emp1.employee_id AS Emp_Santa_ID,
79     emp1.first_name AS First_Name_Santa,
80     emp1.last_name AS Last_Name_Santa,
81     emp2.employee_id AS Emp_ID,
82     emp2.first_name AS First_Name,
83     emp2.last_name AS Last_Name
84 FROM
85     parks_and_recreation.employee_salary AS emp1
86 JOIN
87     parks_and_recreation.employee_salary AS emp2
88 ON
89     emp1.employee_id + 1 = emp2.employee_id
90 ;
```

Emp_Santa_ID	First_Name_Santa	Last_Name_Santa	Emp_ID	First_Name	Last_Name
1	Leslie	Knope	2	Ron	Swanson
2	Ron	Swanson	3	Tom	Haverford
3	Tom	Haverford	4	April	Ludgate
4	April	Ludgate	5	Jerry	Gergich
5	Jerry	Gergich	6	Donna	Meagle
6	Donna	Meagle	7	Ann	Perkins
7	Ann	Perkins	8	Chris	Traeger
8	Chris	Traeger	9	Ben	Wyatt
9	Ben	Wyatt	10	Andy	Dwyer
10	Andy	Dwyer	11	Mark	Brendanawicz
11	Mark	Brendanawicz	12	Craig	Middlebrooks

```
92 # --Query 08 (sampling of joining multiple tables in MySQL)
```

```
93 • SELECT
94     *
95 FROM
96     parks_and_recreation.employee_demographics AS dem
97 INNER JOIN
98     parks_and_recreation.employee_salary AS sal
99 ON
100     dem.employee_id = sal.employee_id
101 INNER JOIN
102     parks_and_recreation.parks_departments AS dept
103 ON
104     sal.dept_id = dept.department_id
105 ;
```

employee_id	first_name	last_name	age	gender	birth_date	employee_id	first_name	last_name	occupation	salary	dept_id	department_id	department_name
1	Leslie	Knope	44	Female	1979-09-25	1	Leslie	Knope	Deputy Director of Parks and Recreation	75000	1	1	Parks and Recreation
3	Tom	Haverford	36	Male	1987-03-04	3	Tom	Haverford	Entrepreneur	50000	1	1	Parks and Recreation
4	April	Ludgate	29	Female	1994-03-27	4	April	Ludgate	Assistant to the Director of Parks and Recreation	25000	1	1	Parks and Recreation
5	Jerry	Gergich	61	Male	1962-08-28	5	Jerry	Gergich	Office Manager	50000	1	1	Parks and Recreation
6	Donna	Meagle	46	Female	1977-07-30	6	Donna	Meagle	Office Manager	60000	1	1	Parks and Recreation
7	Ann	Perkins	35	Female	1988-12-01	7	Ann	Perkins	Nurse	55000	4	4	Healthcare
8	Chris	Traeger	43	Male	1980-11-11	8	Chris	Traeger	City Manager	90000	3	3	Public Works
9	Ben	Wyatt	38	Male	1985-07-26	9	Ben	Wyatt	State Auditor	70000	6	6	Finance
11	Mark	Brendanawicz	40	Male	1983-06-14	11	Mark	Brendanawicz	City Planner	57000	3	3	Public Works
12	Craig	Middlebrooks	37	Male	1986-07-27	12	Craig	Middlebrooks	Parks Director	65000	1	1	Parks and Recreation

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	First_Name	Last_Name	Is_In_Department_Of
▶	Leslie	Knope	Parks and Recreation
	Tom	Haverford	Parks and Recreation
	April	Ludgate	Parks and Recreation
	Jerry	Gergich	Parks and Recreation
	Donna	Meagle	Parks and Recreation
	Ann	Perkins	Healthcare
	Chris	Traeger	Public Works
	Ben	Wyatt	Finance
	Mark	Brendanawicz	Public Works
	Craig	Middlebrooks	Parks and Recreation

5. UNION in MySQL (flavors of queries and its sampling output)

The screenshot shows a MySQL IDE window titled "02-UNION in MySQL". The SQL editor contains the following code:

```

1  #This is selecting value(s) to display from a table, using:
2  #      (a) UNION in MySQL
3
4  # --Query 01 (sampling of union in MySQL)
5  • SELECT
6      first_name,
7      last_name
8  FROM
9      employee_demographics
10 UNION # by default, this is UNION DISTINCT
11 SELECT
12     first_name,
13     last_name
14 FROM
15     employee_salary
16 ;
17

```

Below the editor, the "Result Grid" tab is active, displaying the results of the query. The grid has two columns: "first_name" and "last_name". The results are as follows:

	first_name	last_name
▶	Leslie	Knope
	Tom	Haverford
	April	Ludgate
	Jerry	Gergich
	Donna	Meagle
	Ann	Perkins
	Chris	Traeger
	Ben	Wyatt
	Andy	Dwyer
	Mark	Brendanawicz
	Craig	Middlebrooks
	Ron	Swanson


```
18 # --Query 02 (sampling of union in MySQL)
```

```
19 • SELECT
20     first_name,
21     last_name, 'from demographics table' AS Tag
22 FROM
23     employee_demographics
24 UNION ALL
25 SELECT
26     first_name,
27     last_name, 'from salary table' AS Tag
28 FROM
29     employee_salary
30
31
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	first_name	last_name	Tag
▶	Leslie	Knope	from demographics table
	Tom	Haverford	from demographics table
	April	Ludgate	from demographics table
	Jerry	Gergich	from demographics table
	Donna	Meagle	from demographics table
	Ann	Perkins	from demographics table
	Chris	Traeger	from demographics table
	Ben	Wyatt	from demographics table
	Andy	Dwyer	from demographics table
	Mark	Brendanawicz	from demographics table
	Craig	Middlebrooks	from demographics table
	Leslie	Knope	from salary table
	Ron	Swanson	from salary table
	Tom	Haverford	from salary table
	April	Ludgate	from salary table
	Jerry	Gergich	from salary table
	Donna	Meagle	from salary table
	Ann	Perkins	from salary table
	Chris	Traeger	from salary table
	Ben	Wyatt	from salary table
	Andy	Dwyer	from salary table
	Mark	Brendanawicz	from salary table
	Craig	Middlebrooks	from salary table

```
32 # --Query 03 (sampling of union in MySQL)
```

```
33 # Scenario is list of employee that is equal or more than 50 yrs old or got a salary of more than 70k
```

```
34 • SELECT
35     dem.first_name,
36     dem.last_name, 'Employee in 50s' AS Remarks
37 FROM
38     employee_demographics as dem
39 WHERE
40     age >= 50
41 UNION
42 SELECT
43     sal.first_name,
44     sal.last_name, 'With high salary' AS Remarks
45 FROM
46     employee_salary as sal
47 WHERE
48     salary > 70000
49
50
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	first_name	last_name	Remarks
▶	Jerry	Gergich	Employee in 50s
	Leslie	Knope	With high salary
	Chris	Traeger	With high salary

```

51  # --Query 04 (sampling of union in MySQL)
52  # Scenario is list of employee that is more than 40 yrs old (and their respective gender) or got a salary of more than 70k
53  • SELECT
54      dem.first_name,
55      dem.last_name, 'Employee in 40s and Male' AS Remarks
56  FROM
57      employee_demographics as dem
58  WHERE
59      age > 40
60  AND
61      gender = 'Male'
62  UNION
63  SELECT
64      dem.first_name,
65      dem.last_name, 'Employee in 40s and Female' AS Remarks
66  FROM
67      employee_demographics as dem
68  WHERE
69      age > 40
70  AND
71      gender = 'Female'
72  UNION
73  SELECT
74      sal.first_name,
75      sal.last_name, 'With high salary' AS Remarks
76  FROM
77      employee_salary as sal
78  WHERE
79      salary > 70000
80  ORDER BY
81      first_name,
82      last_name
83

```

Result Grid			
Filter Rows:			
Export:			
Wrap Cell Content:			
first_name	last_name	Remarks	
Chris	Traeger	Employee in 40s and Male	
Chris	Traeger	With high salary	
Donna	Meagle	Employee in 40s and Female	
Jerry	Gergich	Employee in 40s and Male	
Leslie	Knope	Employee in 40s and Female	
Leslie	Knope	With high salary	

6. String Functions in MySQL (flavors of queries and its sampling output)

```

03-String Functions in MySQL x
1  #This is selecting value(s) to display from a table, using:
2  #      (a) String functions in MySQL
3

```

```

4      # --Query 01 (sampling of string function in MySQL)
5      • SELECT LENGTH ('mississippi') AS 'Word Length';
6

```


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

	Word Length
▶	11

```

7      # --Query 02 (sampling of string function in MySQL)
8      • SELECT
9          first_name,
10         LENGTH(first_name) AS Name_length_count
11     FROM
12         parks_and_recreation.employee_demographics
13     ORDER BY 2
14

```


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

	first_name	Name_length_count
▶	Tom	3
	Ann	3
	Ben	3
	Andy	4
	Mark	4
	April	5
	Jerry	5
	Donna	5
	Chris	5
	Craig	5
	Leslie	6





```

16     # --Query 03 (sampling of string function in MySQL)
17     • SELECT UPPER ('mississippi') AS Word_case;
18     • SELECT LOWER ('MISSISSIPPI') AS Word_case;
19

```

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

	Word_case
▶	MISSISSIPPI





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

	Word_case
▶	mississippi

```

20 # --Query 04 (sampling of string function in MySQL)
21 • SELECT
22     first_name,
23     UPPER(first_name) AS Name_case
24 FROM
25     parks_and_recreation.employee_demographics
26 ORDER BY 2
27
28

```



Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	first_name	Name_case
▶	Andy	ANDY
	Ann	ANN
	April	APRIL
	Ben	BEN
	Chris	CHRIS
	Craig	CRAIG
	Donna	DONNA
	Jerry	JERRY
	Leslie	LESLIE
	Mark	MARK
	Tom	TOM




```

29 # --Query 05 (sampling of string function in MySQL)
30 • SELECT TRIM('    space    ') AS Get_rid_of_spaces_from_both_side;
31 • SELECT LTRIM('    space    ') AS Get_rid_of_spaces_from_left_side;
32 • SELECT RTRIM('    space    ') AS Get_rid_of_spaces_from_right_side;
33

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Get_rid_of_spaces_from_both_side
▶	space

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Get_rid_of_spaces_from_left_side
▶	space

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Get_rid_of_spaces_from_right_side
▶	space

```

34 # --Query 06 (sampling of string function in MySQL)
35 • SELECT
36     first_name,
37     LEFT(first_name, 4),
38     RIGHT(first_name, 4)
39 FROM
40     parks_and_recreation.employee_demographics
41 ;
42

```



Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	first_name	LEFT(first_name, 4)	RIGHT(first_name, 4)
▶	Leslie	Lesl	slie
	Tom	Tom	Tom
	April	Apri	pril
	Jerry	Jerr	erry
	Donna	Donn	onna
	Ann	Ann	Ann
	Chris	Chri	hris
	Ben	Ben	Ben
	Andy	Andy	Andy
	Mark	Mark	Mark
	Craig	Crai	raig

```

43 # --Query 07 (sampling of string function in MySQL)
44 • SELECT
45     first_name,
46     LEFT(first_name, 4),
47     RIGHT(first_name, 4),
48     SUBSTRING(first_name, 3,2),
49     birth_date,
50     SUBSTRING(birth_date,6,2) AS Birth_month
51 FROM
52     parks_and_recreation.employee_demographics
53 ;
54

```

Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	first_name	LEFT(first_name, 4)	RIGHT(first_name, 4)	SUBSTRING(first_name, 3,2)	birth_date	Birth_month
▶	Leslie	Lesl	slie	sl	1979-09-25	09
	Tom	Tom	Tom	m	1987-03-04	03
	April	Apri	pril	ri	1994-03-27	03
	Jerry	Jerr	erry	rr	1962-08-28	08
	Donna	Donn	onna	nn	1977-07-30	07
	Ann	Ann	Ann	n	1988-12-01	12
	Chris	Chri	hris	ri	1980-11-11	11
	Ben	Ben	Ben	n	1985-07-26	07
	Andy	Andy	Andy	dy	1989-03-25	03
	Mark	Mark	Mark	rk	1983-06-14	06
	Craig	Crai	raig	ai	1986-07-27	07

```
55 # --Query 08 (sampling of string function in MySQL)
```


```
56 • SELECT
57     first_name,
58     REPLACE(first_name, 'a', 'ZZ')
59 FROM
60     parks_and_recreation.employee_demographics
61 ;
62
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 



	first_name	REPLACE(first_name, 'a', 'ZZ')
▶	Leslie	Leslie
	Tom	Tom
	April	April
	Jerry	Jerry
	Donna	DonnZZ
	Ann	Ann
	Chris	Chris
	Ben	Ben
	Andy	Andy
	Mark	MZZrk
	Craig	CrZZig

```
63 # --Query 09 (sampling of string function in MySQL)
```

```
64 • SELECT LOCATE('e','Lopez');
65
66 • SELECT
67     first_name,
68     LOCATE('An', first_name)
69 FROM
70     parks_and_recreation.employee_demographics
71 ;
72
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	LOCATE('e','Lopez')
▶	4

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	first_name	LOCATE('An', first_name)
▶	Leslie	0
	Tom	0
	April	0
	Jerry	0
	Donna	0
	Ann	1
	Chris	0
	Ben	0
	Andy	1
	Mark	0
	Craig	0

```

73 # --Query 10 (sampling of string function in MySQL)
74 • SELECT
75     first_name,
76     last_name,
77     CONCAT(last_name, ', ', first_name) AS Full_name
78 FROM
79     parks_and_recreation.employee_demographics
80 ;

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	first_name	Last_name	Full_name
▶	Leslie	Knope	Knope, Leslie
	Tom	Haverford	Haverford, Tom
	April	Ludgate	Ludgate, April
	Jerry	Gergich	Gergich, Jerry
	Donna	Meagle	Meagle, Donna
	Ann	Perkins	Perkins, Ann
	Chris	Traeger	Traeger, Chris
	Ben	Wyatt	Wyatt, Ben
	Andy	Dwyer	Dwyer, Andy
	Mark	Brendanawicz	Brendanawicz, Mark
	Craig	Middlebrooks	Middlebrooks, Craig

7. CASE Statements in MySQL (flavors of queries and its sampling output)

```

04-CASE Statements in MySQL
Limit to 1000 rows

1 #This is selecting value(s) to display from a table, using:
2 # (a) CASE statements in MySQL
3
4 # --Query 01 (sampling of case statement in MySQL)
5 # Scenario is list of employee grouped by age criteria
6 • SELECT
7     first_name,
8     last_name,
9     age,
10    CASE
11        WHEN age <=40 THEN 'YOUNG'
12        WHEN age BETWEEN 41 AND 50 THEN 'YOUNG AT HEART'
13        WHEN age >= 51 THEN 'OLD'
14    END AS 'Age Criteria'
15 FROM
16     parks_and_recreation.employee_demographics
17 ;
18

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	first_name	last_name	age	Age Criteria
▶	Leslie	Knope	44	YOUNG AT HEART
	Tom	Haverford	36	YOUNG
	April	Ludgate	29	YOUNG
	Jerry	Gergich	61	OLD
	Donna	Meagle	46	YOUNG AT HEART
	Ann	Perkins	35	YOUNG
	Chris	Traeger	43	YOUNG AT HEART
	Ben	Wyatt	38	YOUNG
	Andy	Dwyer	34	YOUNG
	Mark	Brendanawicz	40	YOUNG
	Craig	Middlebrooks	37	YOUNG

```

19 # --Query 02 (sampling of case statement in MySQL)
20 # Scenario is list of employee grouped by their respective pay increase and bonus
21 # Requirements:
22 # (a) < 50000 gets a 5% bonus,
23 # (b) > 50000 gets a 7% bonus, and
24 # (c) any who is in Finance gets an extra 10% bonus
25 • SELECT
26     first_name,
27     last_name,
28     salary,
29     CASE
30         WHEN salary < 50000 THEN salary + (salary * 0.05)
31         WHEN salary > 50000 THEN salary + (salary * 0.07)
32     END AS 'New Salary',
33     CASE
34         WHEN dept_id = 6 THEN salary * 0.10 # this is for folks in the Finance Dept
35     END AS 'New Salary (Finance Group)'
36 FROM
37     parks_and_recreation.employee_salary
38

```

Result Grid					
		Filter Rows:		Export:	Wrap Cell Content:
	first_name	last_name	salary	New Salary	New Salary (Finance Group)
▶	Leslie	Knope	75000	80250.00	NULL
	Ron	Swanson	70000	74900.00	NULL
	Tom	Haverford	50000	NULL	NULL
	April	Ludgate	25000	26250.00	NULL
	Jerry	Gergich	50000	NULL	NULL
	Donna	Meagle	60000	64200.00	NULL
	Ann	Perkins	55000	58850.00	NULL
	Chris	Traeger	90000	96300.00	NULL
	Ben	Wyatt	70000	74900.00	7000.00
	Andy	Dwyer	20000	21000.00	NULL
	Mark	Brendanawicz	57000	60990.00	NULL
	Craig	Middlebrooks	65000	69550.00	NULL

8. Subqueries in MySQL (flavors of queries and its sampling output)

```

05-Subqueries in MySQL x
1 #This is selecting value(s) to display from a table, using:
2 # (a) Subqueries in MySQL
3

```



```

4  # --Query 01 (sampling of subquery in MySQL)
5  # Scenario is only display employees that works for Parks and Recreation department (at WHERE level)
6  • SELECT
7      *
8  FROM
9      parks_and_recreation.employee_demographics
10 WHERE
11     employee_id IN (
12         SELECT
13             employee_id
14         FROM
15             employee_salary
16         WHERE
17             dept_id = 1
18     )
19 ;
20

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	employee_id	first_name	last_name	age	gender	birth_date
▶	1	Leslie	Knope	44	Female	1979-09-25
	3	Tom	Haverford	36	Male	1987-03-04
	4	April	Ludgate	29	Female	1994-03-27
	5	Jerry	Gergich	61	Male	1962-08-28
	6	Donna	Meagle	46	Female	1977-07-30
	12	Craig	Middlebrooks	37	Male	1986-07-27
*	NULL	NULL	NULL	NULL	NULL	NULL

```

21 # --Query 02 (sampling of subquery in MySQL)
22 # Scenario is to compare individual employee salary to the total average salary (at SELECT level)
23 • SELECT
24     CONCAT(last_name, ', ', first_name) AS 'Full Name',
25     salary AS 'Employee Salary',
26     (SELECT
27         AVG(salary)
28     FROM
29         employee_salary
30     ) AS 'Total Average Salary'
31 FROM
32     employee_salary
33 ;
34

```

Result Grid

Filter Rows:

Export:




Wrap Cell Content:

	Full Name	Employee Salary	Total Average Salary
▶	Knope, Leslie	75000	57250.0000
	Swanson, Ron	70000	57250.0000
	Haverford, Tom	50000	57250.0000
	Ludgate, April	25000	57250.0000
	Gergich, Jerry	50000	57250.0000
	Meagle, Donna	60000	57250.0000
	Perkins, Ann	55000	57250.0000
	Traeger, Chris	90000	57250.0000
	Wyatt, Ben	70000	57250.0000
	Dwyer, Andy	20000	57250.0000
	Brendanawicz, Mark	57000	57250.0000
	Middlebrooks, Craig	65000	57250.0000

```

35 # --Query 03 (sampling of subquery in MySQL)
36 # Scenario is to get average age, maximum age, minimum age, and total count by gender
37 # Then get average maximum age, average minimum age (at FROM level)
38 • SELECT
39     gender AS 'Gender',
40     AVG(age) AS 'Average Age',
41     MAX(age) AS 'Maximum Age',
42     MIN(age) AS 'Minimum Age',
43     COUNT(age) AS 'Total by gender'
44 FROM
45     parks_and_recreation.employee_demographics
46 GROUP BY
47     gender
48 ;

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Gender	Average Age	Maximum Age	Minimum Age	Total by gender
▶	Female	38.5000	46	29	4
	Male	41.2857	61	34	7

```

49 # Leveraging the above query...
50 • SELECT
51     gender,
52     Average_Age,
53     AVG(Maximum_Age) AS 'Average Max Age',
54     AVG(Minimum_Age) AS 'Average Min Age',
55     Total_by_gender
56 FROM (
57     SELECT
58         gender AS 'Gender',
59         AVG(age) AS 'Average_Age',
60         MAX(age) AS 'Maximum_Age',
61         MIN(age) AS 'Minimum_Age',
62         COUNT(age) AS 'Total_by_gender'
63     FROM
64         parks_and_recreation.employee_demographics
65     GROUP BY gender
66 ) AS Temp_Table
67 GROUP BY gender
68 ;

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	gender	Average_Age	Average Max Age	Average Min Age	Total_by_gender
▶	Female	38.5000	46.0000	29.0000	4
	Male	41.2857	61.0000	34.0000	7

9. Window Functions in MySQL (flavors of queries and its sampling output)

06-Window Functions in MySQL

Limit to 1000 rows

```
1 #This is selecting value(s) to display from a table, using:
2 # (a) Window Function in MySQL
3
4 # --Query 01 (sampling of window function in MySQL)
5 # Query using GROUP BY (aggregate function of AVERAGE)
6 • SELECT
7     gender,
8     AVG(salary) AS 'Average salary by gender'
9 FROM
10    parks_and_recreation.employee_demographics AS dem
11 JOIN
12    parks_and_recreation.employee_salary AS sal
13 ON
14    dem.employee_id = sal.employee_id
15 GROUP BY gender
16 ;
```

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

	gender	Average salary by gender
▶	Female	53750.0000
	Male	57428.5714

```
17 # Query using WINDOW FUNCTION (aggregate function of AVERAGE)
18 • SELECT
19     CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
20     gender AS 'Gender',
21     AVG(salary) OVER(PARTITION BY gender) AS 'Average salary'
22 FROM
23    parks_and_recreation.employee_demographics AS dem
24 JOIN
25    parks_and_recreation.employee_salary AS sal
26 ON
27    dem.employee_id = sal.employee_id
28
29
```

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

	Full Name	Gender	Average salary
▶	Knope, Leslie	Female	53750.0000
	Ludgate, April	Female	53750.0000
	Meagle, Donna	Female	53750.0000
	Perkins, Ann	Female	53750.0000
	Haverford, Tom	Male	57428.5714
	Gergich, Jerry	Male	57428.5714
	Traeger, Chris	Male	57428.5714
	Wyatt, Ben	Male	57428.5714
	Dwyer, Andy	Male	57428.5714
	Brendanawicz, Mark	Male	57428.5714
	Middlebrooks, Craig	Male	57428.5714

```

30 # --Query 02 (sampling of window function in MySQL)
31 # Query using WINDOW FUNCTION (aggregate function of SUM)
32 • SELECT
33     CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
34     gender AS 'Gender',
35     SUM(salary) OVER(PARTITION BY gender) AS 'Average salary'
36 FROM
37     parks_and_recreation.employee_demographics AS dem
38 JOIN
39     parks_and_recreation.employee_salary AS sal
40 ON
41     dem.employee_id = sal.employee_id
42

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Full Name	Gender	Average salary
▶	Knope, Leslie	Female	215000
	Ludgate, April	Female	215000
	Meagle, Donna	Female	215000
	Perkins, Ann	Female	215000
	Haverford, Tom	Male	402000
	Gergich, Jerry	Male	402000
	Traeger, Chris	Male	402000
	Wyatt, Ben	Male	402000
	Dwyer, Andy	Male	402000
	Brendanawicz, Mark	Male	402000
	Middlebrooks, Craig	Male	402000

```

44 # --Query 03 (sampling of window function in MySQL)
45 # Query using WINDOW FUNCTION (rolling total)
46 • SELECT
47     CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
48     gender AS 'Gender',
49     salary AS 'Salary',
50     SUM(salary) OVER(PARTITION BY gender ORDER BY dem.employee_id) AS 'Rolling Total'
51 FROM
52     parks_and_recreation.employee_demographics AS dem
53 JOIN
54     parks_and_recreation.employee_salary AS sal
55 ON
56     dem.employee_id = sal.employee_id
57
58

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Full Name	Gender	Salary	Rolling Total
▶	Knope, Leslie	Female	75000	75000
	Ludgate, April	Female	25000	100000
	Meagle, Donna	Female	60000	160000
	Perkins, Ann	Female	55000	215000
	Haverford, Tom	Male	50000	50000
	Gergich, Jerry	Male	50000	100000
	Traeger, Chris	Male	90000	190000
	Wyatt, Ben	Male	70000	260000
	Dwyer, Andy	Male	20000	280000
	Brendanawicz, Mark	Male	57000	337000
	Middlebrooks, Craig	Male	65000	402000

```

59 # --Query 04 (sampling of window function in MySQL)
60 # Query using WINDOW FUNCTION (ROW NUMBER, RANK, and DENSE RANK)
61 • SELECT
62     dem.employee_id,
63     CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
64     gender AS 'Gender',
65     salary AS 'Salary',
66     ROW_NUMBER() OVER(PARTITION BY gender ORDER BY salary DESC) AS 'Row Number', # will give unique number per row (sort of ranking)
67     RANK() OVER(PARTITION BY gender ORDER BY salary DESC) AS 'Ranking by Gender (positional)', # will give actual rank per row, where ranking is positionally
68     DENSE_RANK() OVER(PARTITION BY gender ORDER BY salary DESC) AS 'Ranking by Gender (numerical)' # will give actual rank per row, where ranking is numerically
69 FROM
70     parks_and_recreation.employee_demographics AS dem
71 JOIN
72     parks_and_recreation.employee_salary AS sal
73 ON
74     dem.employee_id = sal.employee_id
75 ;

```

employee_id	Full Name	Gender	Salary	Row Number	Ranking by Gender (positional)	Ranking by Gender (numerical)
1	Knope, Leslie	Female	75000	1	1	1
6	Meagle, Donna	Female	60000	2	2	2
7	Perkins, Ann	Female	55000	3	3	3
4	Ludgate, April	Female	25000	4	4	4
8	Traeger, Chris	Male	90000	1	1	1
9	Wyatt, Ben	Male	70000	2	2	2
12	Middlebrooks, Craig	Male	65000	3	3	3
11	Brendanawicz, Mark	Male	57000	4	4	4
3	Haverford, Tom	Male	50000	5	5	5
5	Gergich, Jerry	Male	50000	6	5	5
10	Dwyer, Andy	Male	20000	7	7	6

*****END*****