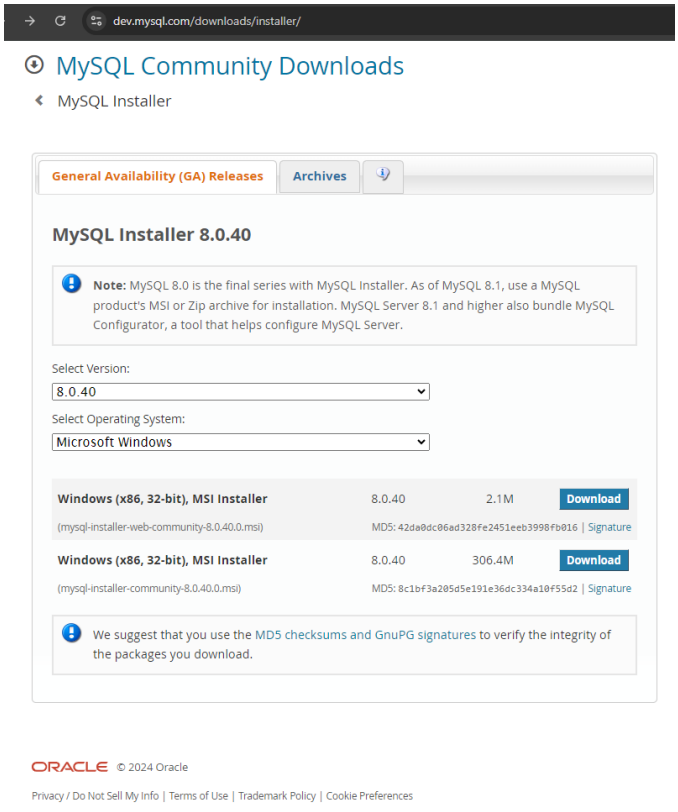
Document highlight:

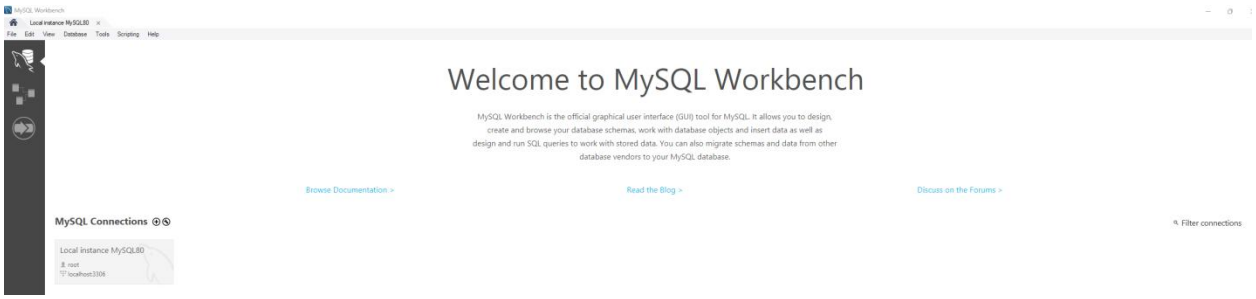
- JOIN in MySQL
- UNION in MySQL
- String Functions in MySQL
    - ie., Length, Upper, Lower, Trim, Left, Right, Substring, Replace, Locate, Concat
- CASE Statements in MySQL
- Subqueries in MySQL
- Window Functions in MySQL
    - ie., Clause - Over, Partition By, Order By
    - ie., Rank, Dense Rank,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

1. Setup MySQL Workbench (community)

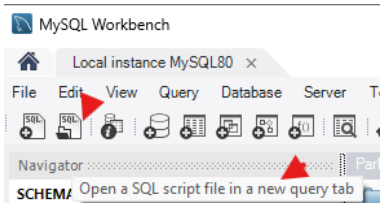
Download: and install (follow installation guide):
https://dev.mysql.com/downloads/installer/



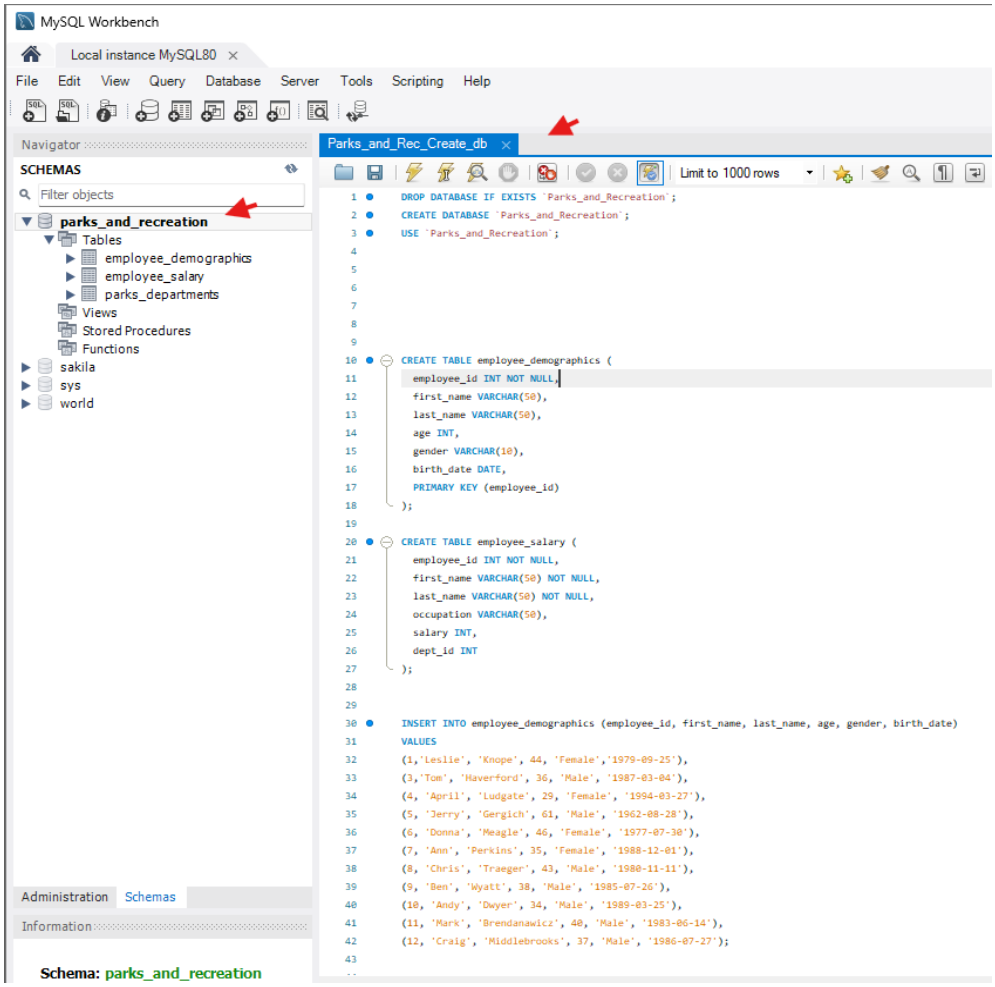
MySQL home screen



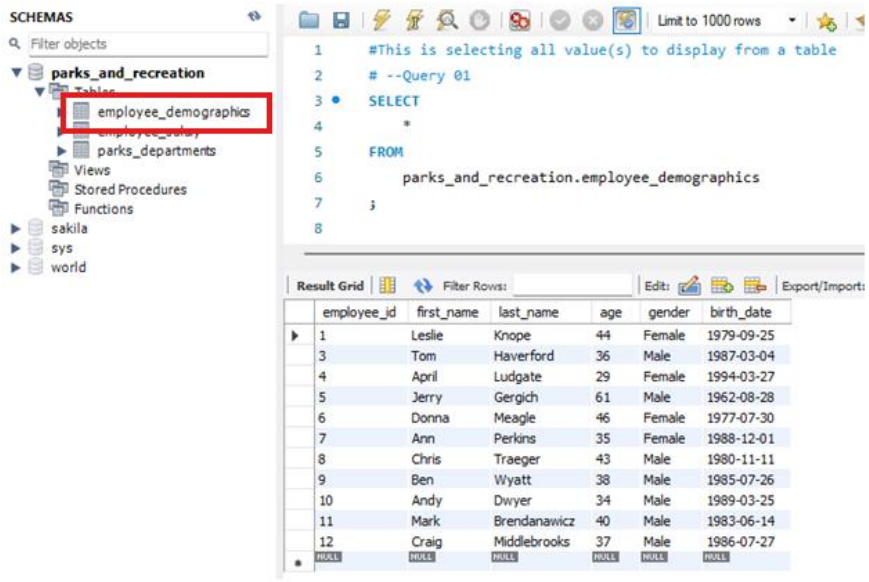
2. Create a database (create tables and load the set of data into it)

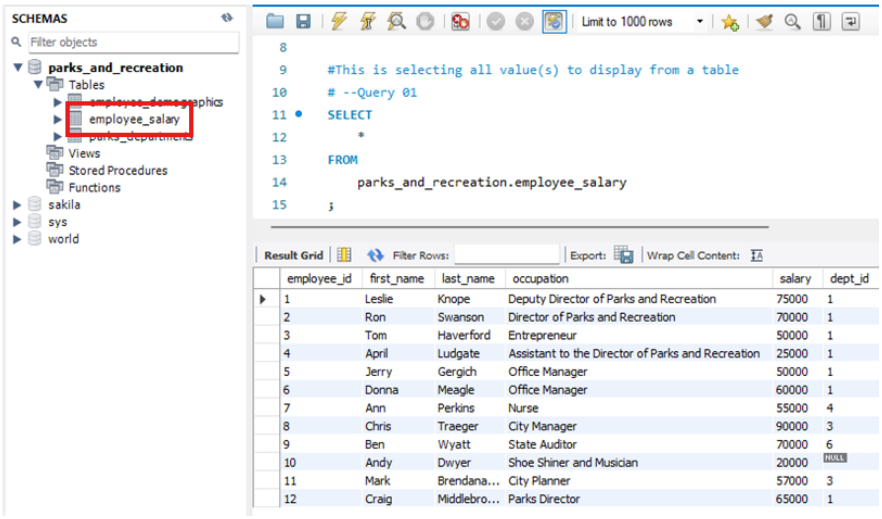


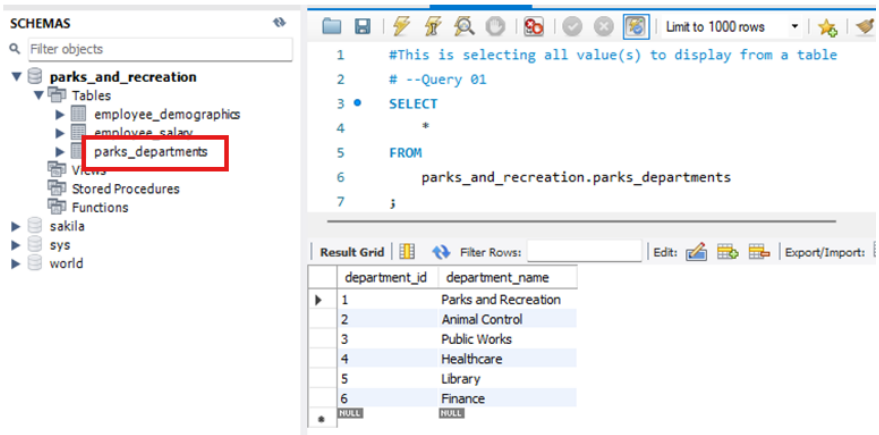
After running the script:



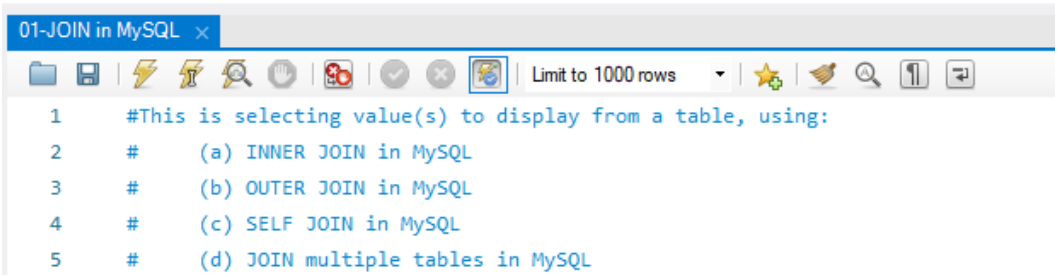
3. Main reference tables







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




```
# --Query 01 (sampling of inner join in MySQL)
SELECT
    *
FROM
    parks_and_recreation.employee_demographics
INNER JOIN
    parks_and_recreation.employee_salary
ON
    employee_demographics.employee_id = employee_salary.employee_id
;
```

Result Grid | Filter Rows: | Export: | 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

```
# --Query 02 (sampling of inner join in MySQL)
SELECT
    *
FROM
    parks_and_recreation.employee_demographics AS dem
INNER JOIN
    parks_and_recreation.employee_salary AS sal
ON
    dem.employee_id = sal.employee_id
;
```

Result Grid | Filter Rows: | Export: | 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

```sql
# --Query 03 (sampling of inner join in MySQL)
SELECT
    dem.employee_id AS ID,
    age AS Age,
    occupation AS Occupation,
    sal.salary AS Salary
FROM
    parks_and_recreation.employee_demographics AS dem
INNER JOIN
    parks_and_recreation.employee_salary AS sal
ON
    dem.employee_id = sal.employee_id
;
```

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

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

```sql
# --Query 04 (sampling of outer join in MySQL)
SELECT
    *
FROM
    parks_and_recreation.employee_demographics AS dem
LEFT OUTER JOIN
    parks_and_recreation.employee_salary AS sal
ON
    dem.employee_id = sal.employee_id
;
```

Result Grid | Filter Rows: | Export: | 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

```sql
# --Query 05 (sampling of outer join in MySQL)
SELECT
    *
FROM
    parks_and_recreation.employee_demographics AS dem
RIGHT OUTER JOIN
    parks_and_recreation.employee_salary AS sal
ON
    dem.employee_id = sal.employee_id
;
```

Result Grid | Filter Rows: | Export: | 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
NULL	NULL	NULL	NULL	NULL	NULL	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	NULL
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

```sql
# --Query 06 (sampling of self join in MySQL)
SELECT
    *
FROM
    parks_and_recreation.employee_salary AS emp1
JOIN
    parks_and_recreation.employee_salary AS emp2
ON
    emp1.employee_id + 1 = emp2.employee_id
;
```

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

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	NULL
10	Andy	Dwyer	Shoe Shiner and Musician	20000	NULL	11	Mark	Brendanawicz	City Planner	57000	3
11	Mark	Brendanawicz	City Planner	57000	3	12	Craig	Middlebrooks	Parks Director	65000	1

```sql
# --Query 07 (sampling of self join in MySQL)
SELECT
    emp1.employee_id AS Emp_Santa_ID,
    emp1.first_name AS First_Name_Santa,
    emp1.last_name AS Last_Name_Santa,
    emp2.employee_id AS Emp_ID,
    emp2.first_name AS First_Name,
    emp2.last_name AS Last_Name
FROM
    parks_and_recreation.employee_salary AS emp1
JOIN
    parks_and_recreation.employee_salary AS emp2
ON
    emp1.employee_id + 1 = emp2.employee_id
;
```

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

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

```sql
# --Query 08 (sampling of joining multiple tables in MySQL)
SELECT
    *
FROM
    parks_and_recreation.employee_demographics AS dem
INNER JOIN
    parks_and_recreation.employee_salary AS sal
ON
    dem.employee_id = sal.employee_id
INNER JOIN
    parks_and_recreation.parks_departments AS dept
ON
    sal.dept_id = dept.department_id
;
```

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

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

```sql
# --Query 09 (sampling of joining multiple tables in MySQL)
SELECT
    dem.first_name AS First_Name,
    dem.last_name AS Last_Name,
    dept.department_name AS Is_In_Department_Of
FROM
    parks_and_recreation.employee_demographics AS dem
INNER JOIN
    parks_and_recreation.employee_salary AS sal
ON
    dem.employee_id = sal.employee_id
INNER JOIN
    parks_and_recreation.parks_departments AS dept
ON
    sal.dept_id = dept.department_id
;
```

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)

02-UNION in MySQL

Limit to 1000 rows

```sql
#This is selecting value(s) to display from a table, using:
#     (a) UNION in MySQL

# --Query 01 (sampling of union in MySQL)
SELECT
    first_name,
    last_name
FROM
    employee_demographics
UNION # by default, this is UNION DISTINCT
SELECT
    first_name,
    last_name
FROM
    employee_salary
;
```

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

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

```sql
# --Query 02 (sampling of union in MySQL)
SELECT
    first_name,
    last_name, 'from demographics table' AS Tag
FROM
    employee_demographics
UNION ALL
SELECT
    first_name,
    last_name, 'from salary table' AS Tag
FROM
    employee_salary
;
```

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

```sql
# --Query 03 (sampling of union in MySQL)
# Scenario is list of employee that is equal or more than 50 yrs old or got a salary of more than 70k
SELECT
    dem.first_name,
    dem.last_name, 'Employee in 50s' AS Remarks
FROM
    employee_demographics as dem
WHERE
    age >= 50
UNION
SELECT
    sal.first_name,
    sal.last_name, 'With high salary' AS Remarks
FROM
    employee_salary as sal
WHERE
    salary > 70000
;
```

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

```sql
# --Query 04 (sampling of union in MySQL)
# Scenario is list of employee that is more than 40 yrs old (and their respective gender) or got a salary of more than 70k
SELECT
    dem.first_name,
    dem.last_name, 'Employee in 40s and Male' AS Remarks
FROM
    employee_demographics as dem
WHERE
    age > 40
AND
    gender = 'Male'
UNION
SELECT
    dem.first_name,
    dem.last_name, 'Employee in 40s and Female' AS Remarks
FROM
    employee_demographics as dem
WHERE
    age > 40
AND
    gender = 'Female'
UNION
SELECT
    sal.first_name,
    sal.last_name, 'With high salary' AS Remarks
FROM
    employee_salary as sal
WHERE
    salary > 70000
ORDER BY
    first_name,
    last_name
;
```

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

Limit to 1000 rows

```sql
#This is selecting value(s) to display from a table, using:
#     (a) String functions in MySQL

```

```sql
# --Query 01 (sampling of string function in MySQL)
SELECT LENGTH ('mississippi') AS 'Word Length';
```

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

Word Length
11

```sql
# --Query 02 (sampling of string function in MySQL)
SELECT
    first_name,
    LENGTH(first_name) AS Name_length_count
FROM
    parks_and_recreation.employee_demographics
ORDER BY 2
;
```

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

```sql
# --Query 03 (sampling of string function in MySQL)
SELECT UPPER ('mississippi') AS Word_case;
SELECT LOWER ('MISSISSIPPI') AS Word_case;
```

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

Word_case
MISSISSIPPI

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

Word_case
mississippi

```sql
# --Query 04 (sampling of string function in MySQL)
SELECT
    first_name,
    UPPER(first_name) AS Name_case
FROM
    parks_and_recreation.employee_demographics
ORDER BY 2
;
```

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

```sql
# --Query 05 (sampling of string function in MySQL)
SELECT TRIM('      space      ') AS Get_rid_of_spaces_from_both_side;
SELECT LTRIM('      space      ') AS Get_rid_of_spaces_from_left_side;
SELECT RTRIM('      space      ')  AS Get_rid_of_spaces_from_right_side;
```

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

```sql
# --Query 06 (sampling of string function in MySQL)
SELECT
    first_name,
    LEFT(first_name, 4),
    RIGHT(first_name, 4)
FROM
    parks_and_recreation.employee_demographics
;
```

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

```sql
# --Query 07 (sampling of string function in MySQL)
SELECT
    first_name,
    LEFT(first_name, 4),
    RIGHT(first_name, 4),
    SUBSTRING(first_name, 3,2),
    birth_date,
    SUBSTRING(birth_date,6,2) AS Birth_month
FROM
    parks_and_recreation.employee_demographics
;
```

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

```sql
# --Query 08 (sampling of string function in MySQL)
SELECT
    first_name,
    REPLACE(first_name, 'a', 'ZZ')
FROM
    parks_and_recreation.employee_demographics
;
```

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

```sql
# --Query 09 (sampling of string function in MySQL)
SELECT LOCATE('e','Lopez');

SELECT
    first_name,
    LOCATE('An', first_name)
FROM
    parks_and_recreation.employee_demographics
;
```

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

```sql
# --Query 10 (sampling of string function in MySQL)
SELECT
    first_name,
    Last_name,
    CONCAT(last_name, ', ', first_name) AS Full_name
FROM
    parks_and_recreation.employee_demographics
;
```

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

```sql
#This is selecting value(s) to display from a table, using:
#     (a) CASE statements in MySQL

# --Query 01 (sampling of case statement in MySQL)
# Scenario is list of employee grouped by age criteria
SELECT
    first_name,
    last_name,
    age,
CASE
    WHEN age <=40 THEN 'YOUNG'
    WHEN age BETWEEN 41 AND 50 THEN 'YOUNG AT HEART'
    WHEN age >= 51 THEN 'OLD'
END AS 'Age Criteria'
FROM
    parks_and_recreation.employee_demographics
;
```

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

```sql
# --Query 02 (sampling of case statement in MySQL)
# Scenario is list of employee grouped by their respective pay increase and bonus
# Requirements:
    # (a) < 50000 gets a 5% bonus,
    # (b) > 50000 gets a 7% bonus, and
    # (c) any who is in Finace gets an extra 10% bonus
SELECT
    first_name,
    last_name,
    salary,
 CASE
    WHEN salary < 50000 THEN salary + (salary * 0.05)
    WHEN salary > 50000 THEN salary + (salary * 0.07)
END AS 'New Salary',
CASE
    WHEN dept_id = 6 THEN salary * 0.10 # this is for folks in the Finance Dept
END AS 'New Salary (Finance Group)'
FROM
    parks_and_recreation.employee_salary
;
```

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

Limit to 1000 rows

```sql
#This is selecting value(s) to display from a table, using:
#     (a) Subqueries in MySQL

```

```sql
# --Query 01 (sampling of subquery in MySQL)
# Scenario is only display employees that works for Parks and Recreation department (at WHERE level)
SELECT
    *
FROM
    parks_and_recreation.employee_demographics
WHERE
    employee_id IN (
        SELECT
            employee_id
        FROM
            employee_salary
        WHERE
            dept_id = 1
    )
;
```

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

```sql
# --Query 02 (sampling of subquery in MySQL)
# Scenario is to compare individual employee salary to the total average salary (at SELECT level)
SELECT
    CONCAT(last_name, ', ', first_name) AS 'Full Name',
    salary AS 'Employee Salary',
        (SELECT
            AVG(salary)
        FROM
            employee_salary
        ) AS 'Total Average Salary'
FROM
    employee_salary
;
```

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

```sql
# --Query 03 (sampling of subquery in MySQL)
# Scenario is to get average age, maximum age, minimun age, and total count by gender
# Then get average maximum age, average minimun age (at FROM level)
SELECT
    gender AS 'Gender',
    AVG(age) AS 'Average Age',
    MAX(age) AS 'Maximum Age',
    MIN(age) AS 'Minimum Age',
    COUNT(age) AS 'Total by gender'
FROM
    parks_and_recreation.employee_demographics
GROUP BY
    gender
;
```

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

```sql
# Leveraging the above query...
SELECT
    gender,
    Average_Age,
    AVG(Maximum_Age) AS 'Average Max Age',
    AVG(Minimum_Age) AS 'Average Min Age',
    Total_by_gender
FROM (
    SELECT
        gender AS 'Gender',
        AVG(age) AS 'Average_Age',
        MAX(age) AS 'Maximum_Age',
        MIN(age) AS 'Minimum_Age',
        COUNT(age) AS 'Total_by_gender'
    FROM
        parks_and_recreation.employee_demographics
    GROUP BY gender
    ) AS Temp_Table
GROUP BY gender
;
```

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

```
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

```sql
# --Query 02 (sampling of window function in MySQL)
# Query using WINDOW FUNCTION (aggregate function of SUM)
SELECT
    CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
    gender AS 'Gender',
    SUM(salary) OVER(PARTITION BY gender) AS 'Average salary'
FROM
    parks_and_recreation.employee_demographics AS dem
JOIN
    parks_and_recreation.employee_salary AS sal
    ON
        dem.employee_id = sal.employee_id
;
```

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

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

```sql
# --Query 03 (sampling of window function in MySQL)
# Query using WINDOW FUNCTION (rolling total)
SELECT
    CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
    gender AS 'Gender',
    salary AS 'Salary',
    SUM(salary) OVER(PARTITION BY gender ORDER BY dem.employee_id) AS 'Rolling Total'
FROM
    parks_and_recreation.employee_demographics AS dem
JOIN
    parks_and_recreation.employee_salary AS sal
    ON
        dem.employee_id = sal.employee_id
;
```

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

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

```sql
# --Query 04 (sampling of window function in MySQL)
# Query using WINDOW FUNCTION (ROW NUMBER, RANK, and DENSE RANK)
SELECT
    dem.employee_id,
    CONCAT(dem.last_name, ', ', dem.first_name) AS 'Full Name',
    gender AS 'Gender',
    salary AS 'Salary',
    ROW_NUMBER() OVER(PARTITION BY gender ORDER BY salary DESC) AS 'Row Number', # will give unique number per row (sort of ranking)
    RANK() OVER(PARTITION BY gender ORDER BY salary DESC) AS 'Ranking by Gender (positional)', # will give actual rank per row, where ranking is positionally
    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
FROM
    parks_and_recreation.employee_demographics AS dem
JOIN
    parks_and_recreation.employee_salary AS sal
    ON
        dem.employee_id = sal.employee_id
;
```

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

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**********