

School of Information and Computer Technology
 Sirindhorn International Institute of Technology
 Thammasat University
 ITS331 Information Technology Laboratory I

Laboratory #7: MySQL Exercises

Fill in an appropriate MySQL statement (not results) to complete each task. The structure of `employee_data` and `employee_per` is given for reference.

employee_data

Field	Type	Null	Key	Default	Extra
<code>emp_id</code>	<code>int(10) unsigned</code>	NO	PRI	NULL	<code>auto_increment</code>
<code>f_name</code>	<code>varchar(20)</code>	YES		NULL	
<code>l_name</code>	<code>varchar(20)</code>	YES		NULL	
<code>title</code>	<code>varchar(30)</code>	YES		NULL	
<code>age</code>	<code>int(11)</code>	YES		NULL	
<code>yos</code>	<code>int(11)</code>	YES		NULL	
<code>salary</code>	<code>int(11)</code>	YES		NULL	
<code>perks</code>	<code>int(11)</code>	YES		NULL	
<code>email</code>	<code>varchar(60)</code>	YES		NULL	

employee_per

Field	Type	Null	Key	Default	Extra
<code>e_id</code>	<code>int(10) unsigned</code>	NO	PRI	NULL	
<code>address</code>	<code>varchar(60)</code>	YES		NULL	
<code>phone</code>	<code>int(11)</code>	YES		NULL	
<code>p_email</code>	<code>varchar(60)</code>	YES		NULL	
<code>birth_date</code>	<code>date</code>	YES		NULL	
<code>sex</code>	<code>enum('M', 'F')</code>	YES		NULL	
<code>m_status</code>	<code>enum('Y', 'N')</code>	YES		NULL	
<code>s_name</code>	<code>varchar(40)</code>	YES		NULL	
<code>children</code>	<code>int(11)</code>	YES		NULL	

1. List all available databases

```
show_databases;
```

2. List all available tables

```
show tables;
```

3. Describe the structure of a table named `student`

```
desc student;
```

4. List salary, perks, and yos (years of service) of all employees

```
select emp_id, salary, perks, yos from employee_data;
```

5. List employee IDs, first name, and last name of all “Marketing Executive” who are older than 28 years old

```
SELECT emp_id, f_name, l_name  
FROM employee_data  
WHERE title = "Marketing Executive"  
AND age >28;
```

6. List the first name and last name of all employees who are neither “Senior Programmer” nor “Multimedia Programmer”.

```
SELECT f_name, l_name  
FROM employee_data  
WHERE title NOT IN ('Senior Programmer', 'Multimedia Programmer')
```

7. List first name, last name, title, and age of the top five oldest employees. Order them in descending order of their ages.

```
SELECT f_name, l_name, title, age  
FROM `employee_data`  
ORDER BY age DESC  
LIMIT 5
```

8. Display the highest salary of employees who are “Programmer”.

```
SELECT max( salary ) FROM `employee_data` WHERE title = 'Programmer'
```

9. Display the first name and last name of the youngest employee.

```
SELECT f_name, l_name  
FROM employee_data  
ORDER BY age  
LIMIT 1
```

10. Display the average YOS (year of service) and the sum of salary for each title. HINT: use GROUP BY.

```
SELECT title, avg( yos ) , sum( salary ) FROM employee_data GROUP BY title
```

11. Count the number of employees for each value of ages. Order the results in descending order of ages.

```
SELECT age, count( * )  
FROM employee_data  
GROUP BY age  
ORDER BY age DESC
```

12. Find the average age of each title. Display only titles whose average ages are more than 30. Order them in descending order of the average ages.

```
SELECT avg( age ) , title
FROM employee_data
GROUP BY title
HAVING avg( age ) >30
ORDER BY avg( age ) DESC
```

13. Display employee ids and birth dates (in full e.g., 13 July 2012) of employees born in and between 1970 and 1972.

```
SELECT e_id, concat( day( birth_date ) , ' ', monthname( birth_date ) , ' ',
    year( birth_date ) ) AS 'Birthday'
FROM `employee_per`
WHERE year( birth_date ) BETWEEN 1970 AND 1972
```

14. Count the number of employees who are married and unmarried. Use GROUP BY.

```
SELECT m_status, count( * ) FROM employee_per GROUP BY m_status
```

15. Count the number of male/female employees who are married and unmarried. Note that there are four cases: married male, married female, unmarried male and unmarried female. Use GROUP BY.

```
SELECT sex, m_status, count(*) FROM employee_per GROUP BY sex, m_status
```

16. Display MySQL version

```
SELECT version();
```

17. Change the last name of employee with ID=3 (Anamika Pandit) to “Sharma”

```
UPDATE employee_data SET l_name = "Sharma" WHERE emp_id =3
```

18. Change the titles of all “Multimedia Programmer” to “Multimedia Specialist”

```
UPDATE employee_data SET title = 'Multimedia Specialist' WHERE title =
'Multimedia Programmer'
```

19. Add the following employee to “employee_data”

First name: Rudolf	Last name: Reindeer
Title: Business Analyst	Age: 34
Years of service: 2	Salary: 95000
Perks: 17000	email: rudolf@bugnet.com

```
INSERT INTO employee_data( f_name, l_name, title, age, yos, salary, perks,
email )
VALUES ('Rudolf', 'Reindeer', 'Business Analyst', 34, 2, 95000, 17000,
'rudolf@bugnet.com')
```

20. Display the first name and last name of all employees born in August. Use table join.

```
SELECT f_name, l_name, birth_date
FROM employee_data, employee_per
WHERE emp_id = e_id
AND month( birth_date ) = 8
```

21. Delete all married employees (s_name is not empty) who do not have any children (children is empty) from the table “employee_per”

```
DELETE FROM employee_per WHERE s_name IS NOT NULL AND children IS NULL
```

22. Delete the table “employee_per” from the database

```
DROP TABLE employee_per
```