

Author : Irfan M.M.M
E/15/138
Subject : Advanced DB Systems
Lab : 01
Topic : Review on SQL

Question numbers and the corresponding queries

Q1

```
DROP DATABASE IF EXISTS company;
```

```
CREATE DATABASE company;
```

```
USE company;
```

```
/* create tables */
```

```
create table employees (  
    emp_no int not null,  
    birth_date date not null,  
    first_name varchar(14) not null,  
    last_name varchar(16),  
    sex ENUM('M','F'),  
    hire_date date not null,  
    primary key (emp_no)  
);
```

```
create table titles (  
    emp_no int not null,  
    title varchar(50) not null,  
    from_date date not null,  
    to_date date not null,  
    primary key (emp_no,title),  
    foreign key (emp_no) references employees(emp_no)  
);
```

```
create table salaries (  
    emp_no int not null,  
    salary int not null,  
    from_date date not null,  
    to_date date not null,  
    primary key (emp_no,from_date,to_date),  
    foreign key (emp_no) references employees(emp_no)  
);
```

```
create table departments (  
    dept_name varchar(30) not null,  
    emp_no int not null,  
    primary key (dept_name,emp_no)  
);
```

```

    dept_no char(4) not null,
    dept_name varchar(40) not null,
    primary key (dept_no)
);

```

```

create table dept_manager (
    emp_no int not null,
    dept_no char(4) not null,
    from_date date not null,
    to_date date not null,
    primary key (dept_no, emp_no),
    foreign key (emp_no) references employees(emp_no),
    foreign key (dept_no) references departments(dept_no)
);

```

```

create table dept_emp (
    emp_no int not null,
    dept_no char(4) not null,
    from_date date not null,
    to_date date not null,
    primary key (emp_no, dept_no),
    foreign key (emp_no) references employees(emp_no),
    foreign key (dept_no) references departments(dept_no)
);

```

/* load data from sql files, can replace the relative path with the absolute path if this script is used, please add the .sql resource files in data folder and run the start sql console outside the data folder, or change these paths to absolute paths*/

```

source ./data/load_employees.sql;
source ./data/load_titles.sql;
source ./data/load_departments.sql;
source ./data/load_dept_emp.sql;
source ./data/load_dept_manager.sql;
source ./data/load_salaries1.sql;
source ./data/load_salaries2.sql;

```

```

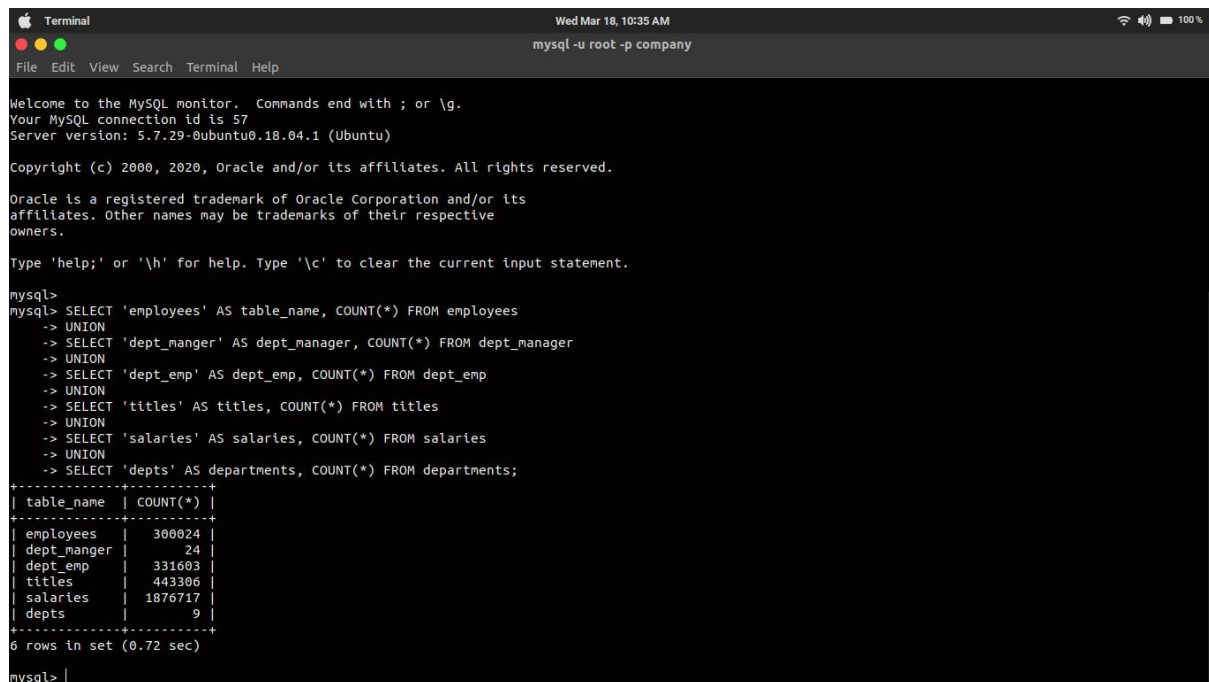
/*-----Q1-----*/
SELECT 'employees' AS table_name, COUNT(*) FROM employees
UNION
SELECT 'dept_manger' AS dept_manager, COUNT(*) FROM dept_manager
UNION

```

```

SELECT 'dept_emp' AS dept_emp, COUNT(*) FROM dept_emp
UNION
SELECT 'titles' AS titles, COUNT(*) FROM titles
UNION
SELECT 'salaries' AS salaries, COUNT(*) FROM salaries
UNION
SELECT 'depts' AS departments, COUNT(*) FROM departments;

```



A terminal window titled 'Terminal' with a dark background. The window shows the MySQL command-line interface. The user has entered a series of UNION SELECT statements to count rows in various tables. The output shows a table with two columns: 'table_name' and 'COUNT(*)'. The rows are: employees (300024), dept_manger (24), dept_emp (331603), titles (443306), salaries (1876717), and depts (9). The terminal also shows the MySQL version (5.7.29-0ubuntu0.18.04.1) and the user (root) connected to the database (company).

```

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 57
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
mysql> SELECT 'employees' AS table_name, COUNT(*) FROM employees
-> UNION
-> SELECT 'dept_manger' AS dept_manager, COUNT(*) FROM dept_manager
-> UNION
-> SELECT 'dept_emp' AS dept_emp, COUNT(*) FROM dept_emp
-> UNION
-> SELECT 'titles' AS titles, COUNT(*) FROM titles
-> UNION
-> SELECT 'salaries' AS salaries, COUNT(*) FROM salaries
-> UNION
-> SELECT 'depts' AS departments, COUNT(*) FROM departments;
+-----+
| table_name | COUNT(*) |
+-----+
| employees | 300024   |
| dept_manger | 24       |
| dept_emp   | 331603   |
| titles     | 443306   |
| salaries   | 1876717  |
| depts      | 9        |
+-----+
6 rows in set (0.72 sec)

mysql>

```

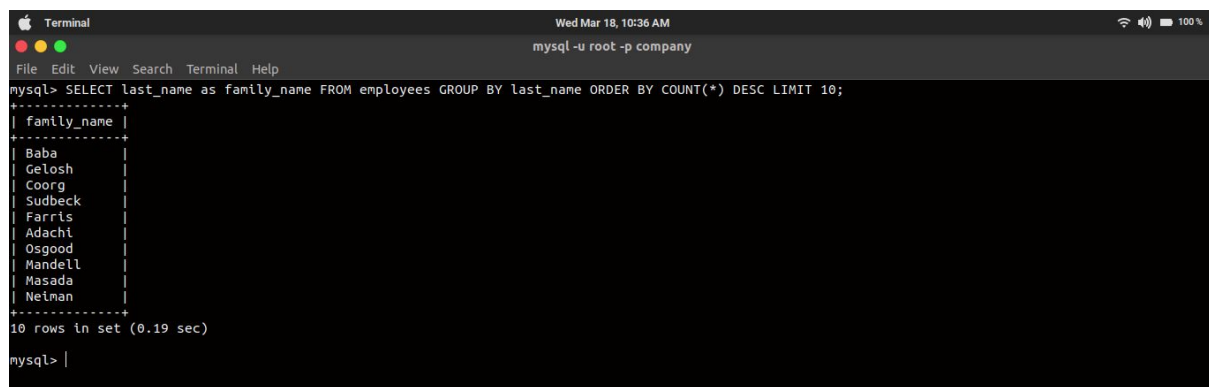
Q2

/*-----Q2-----*/

```

SELECT last_name as family_name FROM employees GROUP BY last_name ORDER
BY COUNT(*) DESC LIMIT 10;

```



A terminal window titled 'Terminal' with a dark background. The user has entered a SQL query to select the last names of employees, grouped by last name, ordered by count descending, and limited to 10 rows. The output shows a table with one column: 'family_name'. The rows are: Baba, Gelosh, Coorg, Sudbeck, Farris, Adachi, Osgood, Mandell, Masada, and Nelman. The terminal also shows the MySQL version (5.7.29-0ubuntu0.18.04.1) and the user (root) connected to the database (company).

```

mysql> SELECT last_name as family_name FROM employees GROUP BY last_name ORDER BY COUNT(*) DESC LIMIT 10;
+-----+
| family_name |
+-----+
| Baba       |
| Gelosh     |
| Coorg      |
| Sudbeck    |
| Farris     |
| Adachi     |
| Osgood     |
| Mandell    |
| Masada     |
| Nelman     |
+-----+
10 rows in set (0.19 sec)

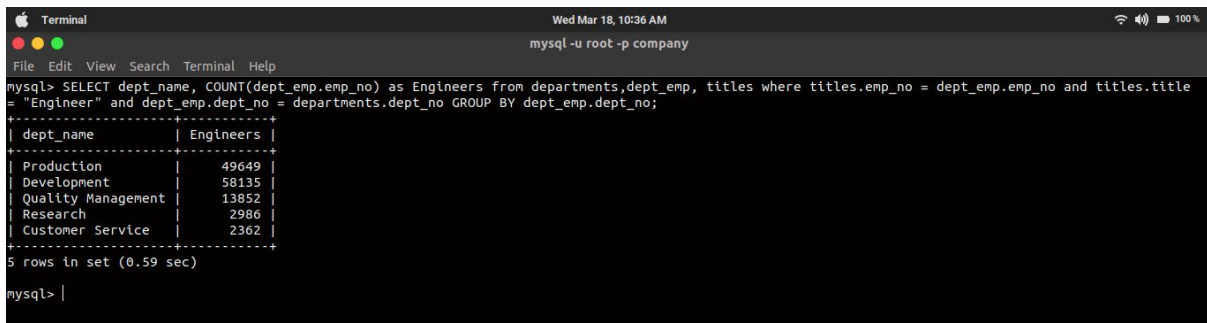
mysql>

```

Q3

```
/*-----Q3-----*/
```

```
SELECT dept_name, COUNT(dept_emp.emp_no) as Engineers from
departments,dept_emp, titles where titles.emp_no = dept_emp.emp_no and
titles.title = "Engineer" and dept_emp.dept_no = departments.dept_no
GROUP BY dept_emp.dept_no;
```



```
mysql> SELECT dept_name, COUNT(dept_emp.emp_no) as Engineers from departments,dept_emp, titles where titles.emp_no = dept_emp.emp_no and titles.title = "Engineer" and dept_emp.dept_no = departments.dept_no GROUP BY dept_emp.dept_no;
```

dept_name	Engineers
Production	49649
Development	58135
Quality Management	13852
Research	2986
Customer Service	2362

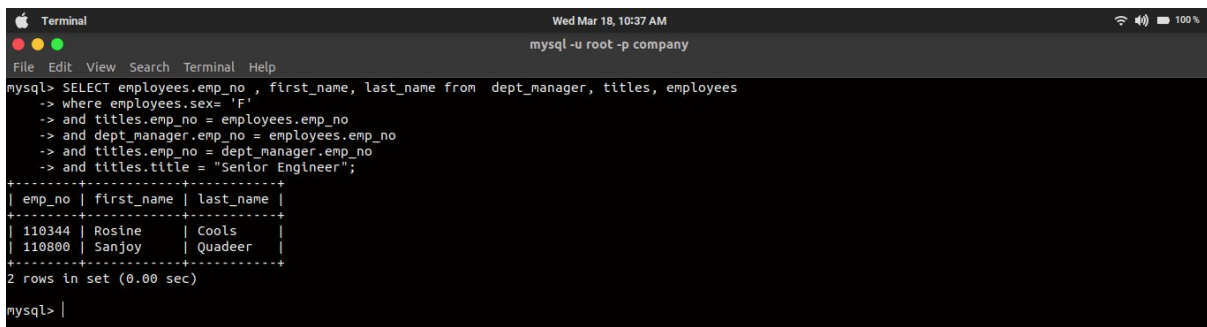
```
5 rows in set (0.59 sec)

mysql>
```

Q4

```
/*-----Q4-----*/
```

```
SELECT employees.emp_no , first_name, last_name from dept_manager,
titles, employees
where employees.sex= 'F'
and titles.emp_no = employees.emp_no
and dept_manager.emp_no = employees.emp_no
and titles.emp_no = dept_manager.emp_no
and titles.title = "Senior Engineer";
```



```
mysql> SELECT employees.emp_no , first_name, last_name from dept_manager, titles, employees
-> where employees.sex= 'F'
-> and titles.emp_no = employees.emp_no
-> and dept_manager.emp_no = employees.emp_no
-> and titles.emp_no = dept_manager.emp_no
-> and titles.title = "Senior Engineer";
```

emp_no	first_name	last_name
110344	Rosine	Cools
110800	Sanjoy	Quadeer

```
2 rows in set (0.00 sec)

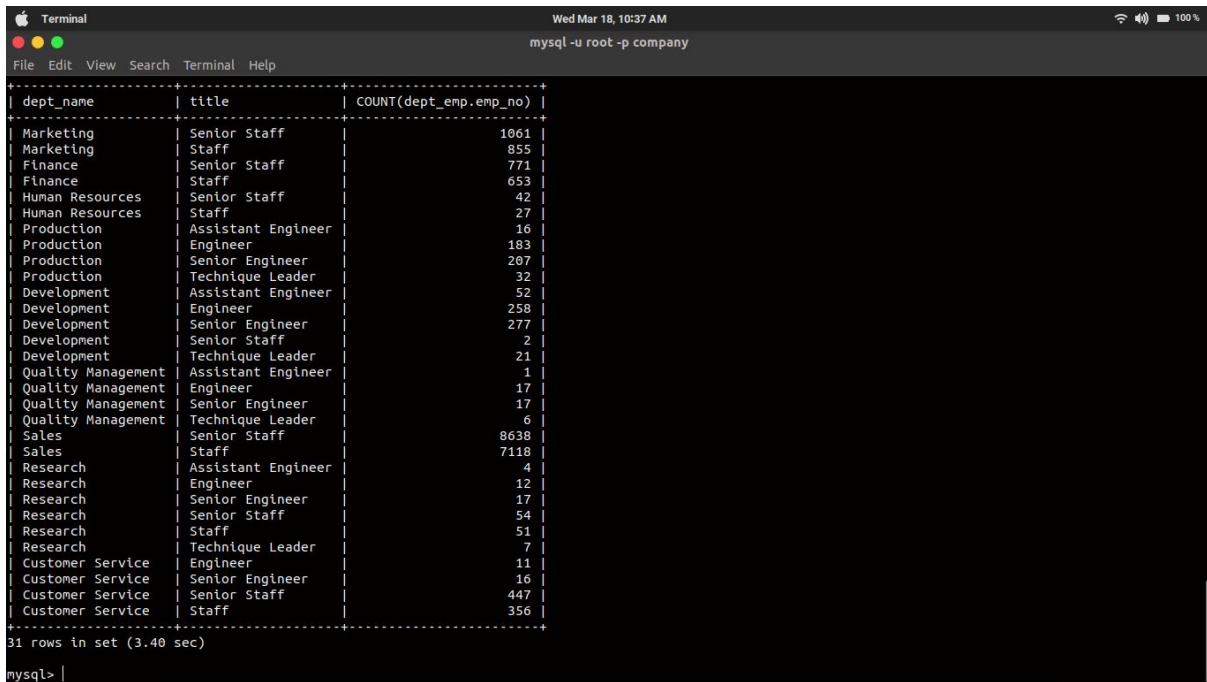
mysql>
```

Q5

```
/*-----Q5-----*/
```

```
SELECT dept_name,titles.title,COUNT(dept_emp.emp_no) from
departments,dept_emp, titles, salaries
where titles.emp_no = dept_emp.emp_no
and dept_emp.dept_no = departments.dept_no
and salaries.salary > 115000
```

```
and salaries.emp_no = dept_emp.emp_no
GROUP BY dept_emp.dept_no , titles.title;
```

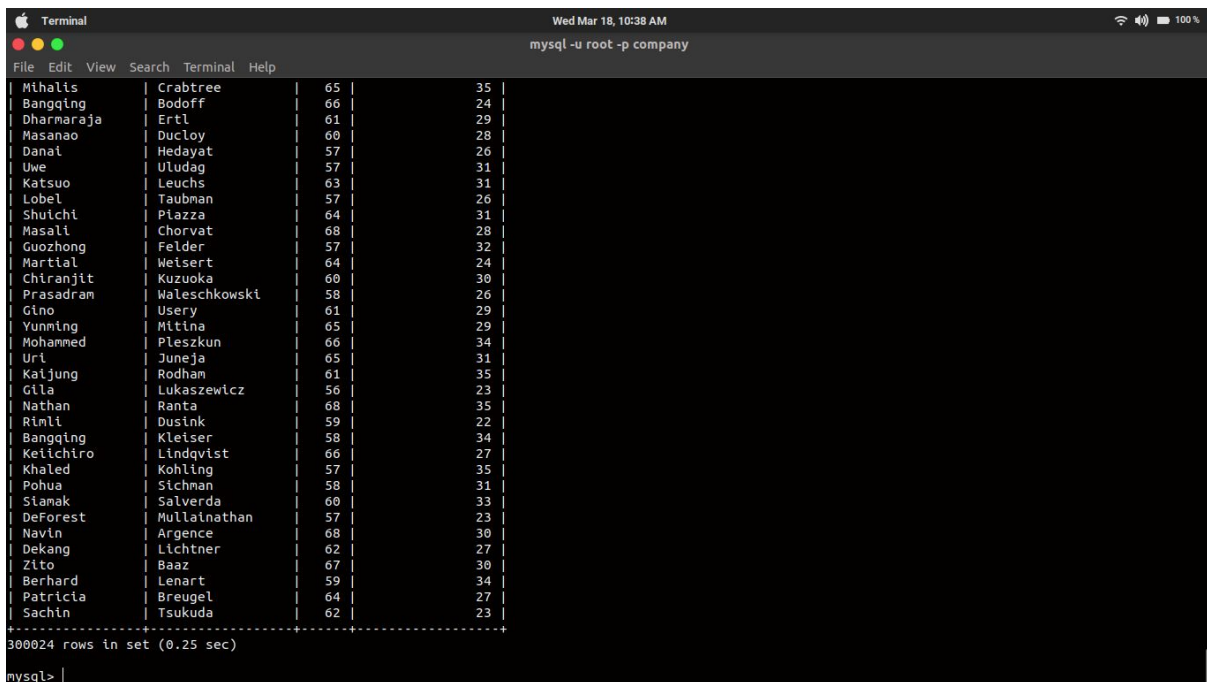


dept_name	title	COUNT(dept_emp.emp_no)
Marketing	Senior Staff	1061
Marketing	Staff	855
Finance	Senior Staff	771
Finance	Staff	653
Human Resources	Senior Staff	42
Human Resources	Staff	27
Production	Assistant Engineer	16
Production	Engineer	183
Production	Senior Engineer	207
Production	Technique Leader	32
Development	Assistant Engineer	52
Development	Engineer	258
Development	Senior Engineer	277
Development	Senior Staff	2
Development	Technique Leader	21
Quality Management	Assistant Engineer	1
Quality Management	Engineer	17
Quality Management	Senior Engineer	17
Quality Management	Technique Leader	6
Sales	Senior Staff	8638
Sales	Staff	7118
Research	Assistant Engineer	4
Research	Engineer	12
Research	Senior Engineer	17
Research	Senior Staff	54
Research	Staff	51
Research	Technique Leader	7
Customer Service	Engineer	11
Customer Service	Senior Engineer	16
Customer Service	Senior Staff	447
Customer Service	Staff	356

31 rows in set (3.40 sec)

Q6

```
/*-----Q6-----*/
SELECT first_name,last_name, (year(CURRENT_DATE())- year(birth_date)) as
age , (year(CURRENT_DATE())- year(hire_date)) as years_of_service
FROM employees
WHERE (year(CURRENT_DATE())- year(birth_date)) > 50
and (year(CURRENT_DATE())- year(hire_date)) > 10;
```



first_name	last_name	age	years_of_service
Mihalis	Crabtree	65	35
Bangqing	Bodoff	66	24
Dharmaraja	Ertl	61	29
Masanao	Ducloy	60	28
Danaei	Hedayat	57	26
Uwe	Uludag	57	31
Katsuo	Leuchs	63	31
Lobel	Taubman	57	26
Shuichi	Piazza	64	31
Masali	Chorvat	68	28
Guozhong	Felder	57	32
Martial	Weisert	64	24
Chiranjit	Kuzuoka	60	30
Prasadram	Waleschkowski	58	26
Gino	Usery	61	29
Yunming	Mitina	65	29
Mohammed	Pleszkun	66	34
Uri	Juneja	65	31
Kaijung	Rodham	61	35
Gila	Lukaszewicz	56	23
Nathan	Ranta	68	35
Rimli	Dusink	59	22
Bangqing	Kleiser	58	34
Keiichiro	Lindqvist	66	27
Khaled	Kohling	57	35
Pohua	Sichman	58	31
Slamak	Salverda	60	33
DeForest	Mullainathan	57	23
Navin	Argence	68	30
Dekang	Lichtner	62	27
Zito	Baaz	67	30
Berhard	Lenart	59	34
Patricia	Bruegel	64	27
Sachin	Tsukuda	62	23

300024 rows in set (0.25 sec)

Q7

```
/*-----Q7-----*/  
SELECT CONCAT(first_name , ' ', last_name) as name from employees  
where emp_no not in (  
SELECT dept_emp.emp_no from dept_emp, departments  
where departments.dept_name = "Human Resources"  
and dept_emp.emp_no = departments.dept_no  
);
```

```
Terminal  
Wed Mar 18, 10:39 AM  
mysql -u root -p company  
File Edit View Search Terminal Help  
Mihalis Crabtree  
Bangqing Bodoff  
Dharmaraja Ertl  
Masanao Ducloy  
Danaï Hedayat  
Uwe Uludag  
Katsuo Leuchs  
Lobel Taubman  
Shuichi Piazza  
Masali Chorvat  
Guozhong Felder  
Martial Weisert  
Chiranjit Kuzuoka  
Prasadram Waleschkowski  
Gino Usery  
Yunming Mittina  
Mohammed Pleszkun  
Uri Juneja  
Kaijung Rodham  
Gila Lukaszewicz  
Nathan Ranta  
Rimil Dusk  
Bangqing Kleiser  
Keiichi Lindqvist  
Khaled Kohling  
Pohua Sichman  
Stamak Salverda  
DeForest Mullainathan  
Navin Argence  
Dekang Lichtner  
Zito Baaz  
Bernhard Lenart  
Patricia Breugel  
Sachin Tsukuda  
-----  
300024 rows in set (0.19 sec)  
mysql> |
```

Q8

```
/*-----Q8-----*/  
SELECT DISTINCT CONCAT(first_name , ' ', last_name) as name from  
employees, salaries  
where employees.emp_no = salaries.emp_no  
and salaries.salary > (  
SELECT max(salary) from salaries, departments, dept_emp  
where departments.dept_name = "Finance"  
and departments.dept_no = dept_emp.dept_no  
and salaries.emp_no = dept_emp.emp_no  
);
```

```
Terminal
Wed Mar 18, 10:39 AM
mysql -u root -p company

File Edit View Search Terminal Help

Nikolaus Businaro
JoAnne Matheson
Wonhee Pagter
Tadanori Sudbeck
Wei Cheng Hatcliff
Chaitali Baik
Mitsuyuki Stanfel
Dines Giaccio
Arnd Junot
Hepling Brender
Sanjai Luders
Honesty Mukaidono
Honglan Otillio
Satoru Gruenwald
Rance Chinin
Eberhardt Gubsky
Wei Jing Chenoweth
Florina Tchuente
Jungsoon Brendel
Shin Birdsall
Mohammed Moehrke
Eldridge Heiserman
Huiqun Birke
Leaf Menyhert
Zhanqiu Muntz
Guenter Ranai
Seongbin Mitsuhashi
Qingxiang Piancastelli
Izaskun Wissmann
Akemi Warwick
Masato Heering
Ghassan Birta
Taiji Malinowski
Lidong Meriste
+-----+
37 rows in set (1.30 sec)

mysql> |
```

Q9

```
/*-----Q9-----*/
SELECT DISTINCT CONCAT(first_name , ' ', last_name) as name from
employees, salaries
where employees.emp_no = salaries.emp_no
and salaries.salary > (
    SELECT AVG(salary) from salaries
);
```

```
Terminal
Wed Mar 18, 10:40 AM
mysql -u root -p company

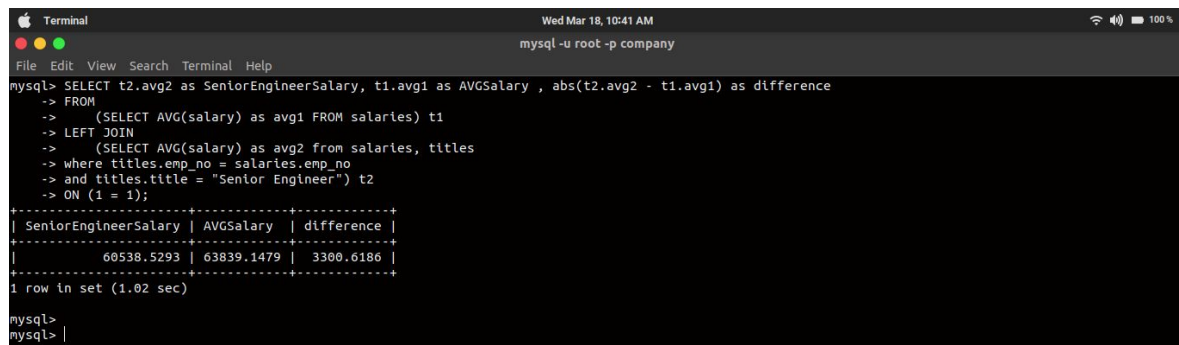
File Edit View Search Terminal Help

Manohar Heemskerk
Moni Bale
Ayakannu Beerel
Remco Rosar
Shigeaki Narlikar
Tonia Butner
Ymte Perelgut
Pantung Litzler
Yolla Auria
Udalprakash Hockney
Krister Stranks
Aron Lindqvist
Leszek Pulkowski
Tuval Tempesti
Kankanahalli Zucker
Lillian Setiz
Gaetan Veldwijk
Holgard Nanard
Danny Lenart
Ronghao Morrow
Mihalis Crabtree
Bangqing Bodoff
Dharmaraja Ertl
Katsuo Leuchs
Guozhong Felder
Martial Weisert
Kaijung Rodham
Nathan Ranta
Bangqing Kleiser
Kellchiro Lindqvist
Slamak Salverda
Zito Baaz
Berhard Lenart
Sachin Tsukuda
+-----+
114280 rows in set (1.67 sec)

mysql> |
```

Q10

```
/*-----Q10-----*/  
SELECT t2.avg2 as SeniorEngineerSalary, t1.avg1 as AVGSalary ,  
abs(t2.avg2 - t1.avg1) as difference  
FROM  
    (SELECT AVG(salary) as avg1 FROM salaries) t1  
LEFT JOIN  
    (SELECT AVG(salary) as avg2 from salaries, titles  
where titles.emp_no = salaries.emp_no  
and titles.title = "Senior Engineer") t2  
ON (1 = 1);
```



```
mysql> SELECT t2.avg2 as SeniorEngineerSalary, t1.avg1 as AVGSalary , abs(t2.avg2 - t1.avg1) as difference  
-> FROM  
->    (SELECT AVG(salary) as avg1 FROM salaries) t1  
-> LEFT JOIN  
->    (SELECT AVG(salary) as avg2 from salaries, titles  
-> where titles.emp_no = salaries.emp_no  
-> and titles.title = "Senior Engineer") t2  
-> ON (1 = 1);  
+-----+  
| SeniorEngineerSalary | AVGSalary | difference |  
+-----+  
| 60538.5293 | 63839.1479 | 3300.6186 |  
+-----+  
1 row in set (1.02 sec)  
  
mysql>  
mysql> |
```

Q11, Q12

```
/*-----Q11. Q12-----*/
```

```
CREATE VIEW current_dept_emp as  
SELECT emp_no, MAX(from_date) as from_date, MAX(to_date) as to_date  
from dept_emp GROUP BY emp_no;
```

```
CREATE VIEW current_dept_emp_with_dept_no as  
SELECT current_dept_emp.emp_no, current_dept_emp.from_date,  
current_dept_emp.to_date, dept_no from current_dept_emp, dept_emp  
where current_dept_emp.emp_no = dept_emp.emp_no  
and current_dept_emp.from_date = dept_emp.from_date  
and current_dept_emp.to_date = dept_emp.to_date;
```

```
SELECT * from current_dept_emp_with_dept_no LIMIT 10;
```



```

Terminal
Wed Mar 18, 10:47 AM
mysql -u root -p company

mysql> CREATE VIEW current_dept_emp as
-> SELECT emp_no, MAX(from_date) as from_date, MAX(to_date) as to_date from dept_emp GROUP BY emp_no;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE VIEW current_dept_emp_with_dept_no as
-> SELECT current_dept_emp.emp_no, current_dept_emp.from_date, current_dept_emp.to_date, dept_no from current_dept_emp, dept_emp
-> where current_dept_emp.emp_no = dept_emp.emp_no
-> and current_dept_emp.from_date = dept_emp.from_date
-> and current_dept_emp.to_date = dept_emp.to_date;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * from current_dept_emp_with_dept_no LIMIT 10;
+-----+
| emp_no | from_date | to_date | dept_no |
+-----+
| 10001 | 1986-06-26 | 9999-01-01 | d005 |
| 10002 | 1996-08-03 | 9999-01-01 | d007 |
| 10003 | 1995-12-03 | 9999-01-01 | d004 |
| 10004 | 1986-12-01 | 9999-01-01 | d004 |
| 10005 | 1989-09-12 | 9999-01-01 | d003 |
| 10006 | 1990-08-05 | 9999-01-01 | d005 |
| 10007 | 1989-02-10 | 9999-01-01 | d008 |
| 10008 | 1998-03-11 | 2000-07-31 | d005 |
| 10009 | 1985-02-18 | 9999-01-01 | d006 |
| 10010 | 2000-06-26 | 9999-01-01 | d006 |
+-----+
10 rows in set (0.44 sec)

mysql> |

```

Q13

/*-----Q13-----*/

```

DROP TRIGGER IF EXISTS `salary_update`;
delimiter //
Create Trigger salary_update
AFTER Update On salaries
For Each Row
Begin
  IF (new.salary > old.salary) Then
    /* print the difference from new.salary - old.salary , but in mysql
    there is no way of printing to the console,it can only throw errors like
    in the other example, or we have to write the changes to a new table
    as of now this throws a warnig*/
    SIGNAL SQLSTATE '01234' set MESSAGE_TEXT = 'salary increment is done';
  End IF;
End;
//
delimiter ;

```

```

Terminal
Wed Mar 18, 11:32 AM
mysql -u root -p company

mysql>
mysql>
mysql> DROP TRIGGER IF EXISTS `salary_update`;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> delimiter //
mysql> Create Trigger salary_update
-> AFTER Update On salaries
-> For Each Row
-> Begin
-> IF (new.salary > old.salary) Then
->   /* print the difference from new.salary - old.salary , but in mysql there is no way of printing to the console,
/* it can only throw errors like in the other example, or we have to write the changes to a new table
/* as of now this throws a warnig
/*> */
->   SIGNAL SQLSTATE '01234' set MESSAGE_TEXT = 'salary increment is done';
-> End IF;
-> End;
-> //
Query OK, 0 rows affected (0.03 sec)

mysql> delimiter ;
mysql> |

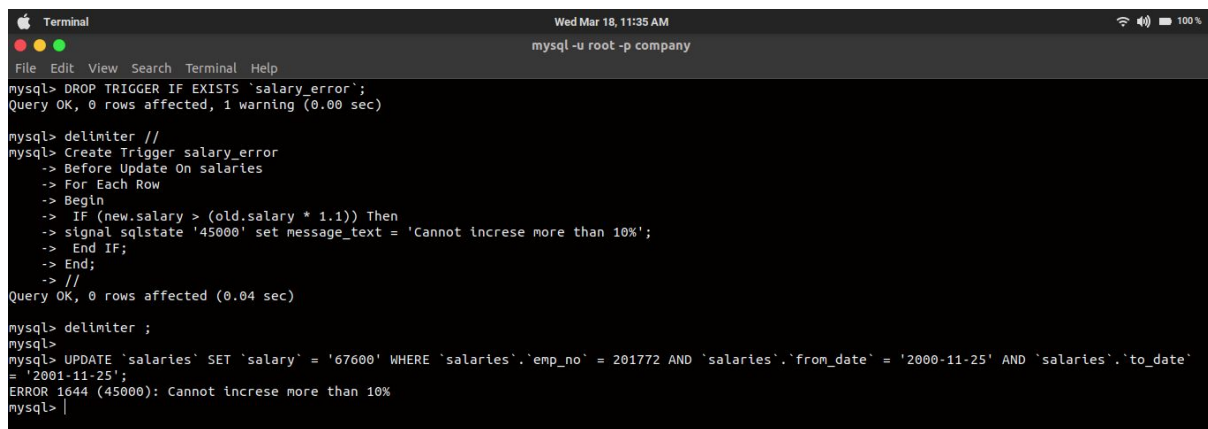
```

Q14

```
/*-----Q14-----*/
```

```
DROP TRIGGER IF EXISTS `salary_error`;  
delimiter //  
Create Trigger salary_error  
Before Update On salaries  
For Each Row  
Begin  
    IF (new.salary > (old.salary * 1.1)) Then  
        signal sqlstate '45000' set message_text = 'Cannot increase more  
than 10%';  
    End IF;  
End;  
//  
delimiter ;
```

```
UPDATE `salaries` SET `salary` = '67600' WHERE `salaries`.`emp_no` =  
201772 AND `salaries`.`from_date` = '2000-11-25' AND  
`salaries`.`to_date` = '2001-11-25';
```



```
Terminal  
Wed Mar 18, 11:35 AM  
mysql -u root -p company  
mysql> DROP TRIGGER IF EXISTS `salary_error`;  
Query OK, 0 rows affected, 1 warning (0.00 sec)  
mysql> delimiter //  
mysql> Create Trigger salary_error  
-> Before Update On salaries  
-> For Each Row  
-> Begin  
-> IF (new.salary > (old.salary * 1.1)) Then  
-> signal sqlstate '45000' set message_text = 'Cannot increase more than 10%';  
-> End IF;  
-> End;  
-> //  
Query OK, 0 rows affected (0.04 sec)  
mysql> delimiter ;  
mysql> UPDATE `salaries` SET `salary` = '67600' WHERE `salaries`.`emp_no` = 201772 AND `salaries`.`from_date` = '2000-11-25' AND `salaries`.`to_date`  
= '2001-11-25';  
ERROR 1644 (45000): Cannot increase more than 10%  
mysql> |
```

NOTE :

- The complete SQL script is in [script.sql](#) file
- The screenshots are in the [screenshots](#) folder
- The add the .sql file resources in the [data](#) folder
(if you wish to load it using the source command since the file size is big, the files a removed.)