January 21, 2015 | Leapfrog Technology Nepal

DB Training Assignment

**Section I**

**question1:** The table given below show the description of EMPLOYEES table.

Employee\_Id int, First\_Name VARCHAR(50), Last\_Name VARCHAR(50), Job\_Id VARCHAR(50), Hire\_Date, Salary Date, Manager\_Id int, Commission\_Percent int, Department\_Id VARCHAR(10)

Look at the above table description and write queries for following questions.

1. Display First name and Job of all employees who do not earn a commission.

2. Display name, job id and data of hiring of all employees who work in IT department

with the most recent dates appearing first.

3. Display no. of employees in each department, alongside department code, only those

employees with salary greater than 3000.

4. Modify Query 3 such that only departments with more than 15 employees are

displayed.

5. Display total salary of each department, alongside department code, with only

departments with total salary > 10000. If any employee has no departments display

“EXECUTIVE”.January 21, 2015 | Leapfrog Technology Nepal

**Answers:**

**Ans-1**

--> SELECT

first\_name, job\_id

FROM

employees

WHERE

commission\_percent IS NULL;

**Ans-2**

-->SELECT

CONCAT\_WS(' ', first\_name, last\_name) AS name,

job\_id,

hire\_date

FROM

employees

WHERE

department\_id = 'IT'

ORDER BY hire\_date DESC;

**Ans-3**

-->SELECT

department\_id, COUNT(\*) AS 'no of employees'

FROM

employees

WHERE

salary > 3000

GROUP BY department\_id;

**Ans-4**

-->SELECT

department\_id, COUNT(\*) AS no\_of\_employees

FROM

employees

WHERE

salary > 3000

GROUP BY department\_id

HAVING no\_of\_employees > 1;

Ans-5

-->SELECT

COALESCE(department\_id, 'EXECUTIVE'),

SUM(salary) AS Total\_Salary

FROM

employees

GROUP BY department\_id

HAVING Total\_Salary > 10000;

<<<<-------Section II------>>>

**Question** 1. Write a query to subtract current date to previous year, but add 2 months.

**Answer**

SELECT

DATE\_ADD(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR),

INTERVAL 2 MONTH);

**Question 2.** Write a query to display output given below using MySQL date functions:

Table Name: tblDATE

|  |  |
| --- | --- |
| Month | Year |
| January | 2013 |
| February | 2013 |
| March | 2013 |

Required Output :

|  |  |  |
| --- | --- | --- |
| Month | Start Date | End Date |
| Jan-13 | 01/01/2013 | 31/01/2013 |
| Feb-13 | 01/02/2013 | 28/02/2013 |
| Mar-13 | 01/03/2013 | 31/03/2013 |

**Answer**

-->SELECT

CONCAT\_WS('-',

SUBSTR(month, 1, 3),

SUBSTR(year, - 2, 2)) AS Month,

CONCAT\_WS('/',

01,

MONTH(STR\_TO\_DATE(SUBSTR(month, 1, 3), '%b')),

year) AS start\_date,

LAST\_DAY(CONCAT\_WS('-',

year,

MONTH(STR\_TO\_DATE(SUBSTR(month, 1, 3), '%b')),

'01')) AS end\_date

FROM

month\_year;

<<<←--------------Section III-------------->>>>

**Questions**

Use Sakila database for all the queries. (Use Subqueries for questions 4, 5 & 6)

1. Select the title and language of all the films with film title starting with ‘A’.

(Tables: film, language)

2. Return the first name, last name and city of all customers who live in Canada. Order

the results first by the last name and then by the first name.

(Tables: customer, address, city, country)

3. Select name of all of the customers who have rented horror movies.

(Tables: customer, rental, inventory, film\_category, category)

4. Select full name of all the actors of films along with the film name, which contain

actor with first name ‘JENNIFER’.

5. Select full name of all of the customers who have rented movies in the category

‘Music’.

6. Show all customer ids and full names who have rented a film in every category with

category\_id >=5.

**Answers**

**Ans 1)**

SELECT

film.title, language.name

FROM

film

JOIN

language ON film.language\_id = language.language\_id

WHERE

title LIKE 'a%';

**Ans 2)**

SELECT

customer.first\_name, customer.last\_name, c.city

FROM

customer

JOIN

address AS a ON customer.address\_id = a.address\_id

JOIN

city AS c ON a.city\_id = c.city\_id

JOIN

country coun ON c.country\_id = coun.country\_id

WHERE

coun.country = 'canada'

ORDER BY customer.first\_name , customer.last\_name;

**Ans 3)**

SELECT

CONCAT\_WS(' ', first\_name, last\_name)

FROM

customer AS c

JOIN

rental AS r ON c.customer\_id = r.customer\_id

JOIN

inventory AS i ON i.inventory\_id = r.inventory\_id

JOIN

film\_category AS fc ON fc.film\_id = i.film\_id

JOIN

category ca ON fc.category\_id = ca.category\_id

WHERE

ca.name = 'horror';

**Ans 4)**

SELECT

CONCAT\_WS(' ', first\_name, last\_name), title

FROM

actor a

JOIN

film\_actor fa ON a.actor\_id = fa.actor\_id

JOIN

film f ON fa.film\_id = f.film\_id

WHERE

f.film\_id IN (SELECT

film\_id

FROM

film\_actor

WHERE

a.actor\_id = (SELECT

actor\_id

FROM

actor

WHERE

first\_name = 'Jennifer'));

**Ans 5)**

SELECT

CONCAT\_WS(' ', first\_name, last\_name)

FROM

customer

WHERE

customer\_id IN (SELECT

customer\_id

FROM

rental

WHERE

inventory\_id IN (SELECT

inventory\_id

FROM

inventory

WHERE

film\_id IN (SELECT

film\_id

FROM

film\_category

WHERE

category\_id IN (SELECT

category\_id

FROM

category

WHERE

name = 'music'))));

**Ans 6)**

SELECT

customer\_id, CONCAT\_WS(' ', first\_name, last\_name)

FROM

customer

WHERE

customer\_id IN (SELECT

customer\_id

FROM

rental

WHERE

inventory\_id IN (SELECT

inventory\_id

FROM

inventory

WHERE

film\_id IN (SELECT

film\_id

FROM

film\_category

WHERE

category\_id IN (SELECT

category\_id

FROM

category

WHERE

category\_id = ALL (SELECT

category\_id

FROM

category

WHERE

category\_id <=5)))));