CO 527 – ADVANCED DATABASE SYSTEMS

LAB 01- REVIEW ON SQL

KARUNACHANDRA R.H.I.O.

E/17/153

1) Load data to each of the tables from the given .sql files.

```
SELECT COUNT(*) FROM employees;
```

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

COUNT(*)
300024

```
SELECT COUNT(*) FROM dept_manager;
```

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

COUNT(*)
24

```
SELECT COUNT(*) FROM dept_emp;
```

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

COUNT(*)
331603

```
SELECT COUNT(*) FROM titles;
```

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

COUNT(*)
443308

```
SELECT COUNT(*) FROM salaries;
```

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

COUNT(*)
1876717

```
SELECT COUNT(*) FROM departments;
```

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

COUNT(*)
9

2) Find the top 10 family names(last_name) in the company.

Query 1

Limit to 1000 rows

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

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

last_name
Baba
Gelosh
Coorg
Farris
Sudbeck
Adachi
Osgood
Mandell
Masada
Neiman

3) List the number of Engineers each department has

Query 1

Limit to 1000 rows

```
SELECT
    dept_name, COUNT(*)
FROM
    titles,departments,employees,dept_emp
WHERE employees.emp_no = titles.emp_no  and dept_emp.dept_no=departments.dept_no and
  dept_emp.emp_no=employees.emp_no and titles.title="Engineer"
GROUP BY dept_name;
```

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

dept_name	COUNT(*)
Customer Service	2362
Development	58135
Production	49649
Quality Management	13852
Research	2986

4) List all the female employees who are department managers and have worked as a senior engineer.

e17153queries*

Limit to 1000 rows

```
select employees.emp_no as ID,concat(first_name," ",last_name) as emp_name
from employees,dept_manager,titles
where employees.emp_no = dept_manager.emp_no
and employees.emp_no = titles.emp_no
and employees.sex="F" and titles.title = "Senior Engineer" ;
```

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

ID	emp_name
110344	Rosine Cools
110800	Sanjoy Quadeer

5) Display the departments and titles of employees who have a salary greater than 115000. Display how many of such employees work for each department.

e17153queries*

Limit to 1000 rows

```
select dept_name,title,Count(*) as total_employees
from salaries,dept_emp,departments,titles
where salaries.emp_no = dept_emp.emp_no and departments.dept_no = dept_emp.dept_no
and titles.emp_no = salaries.emp_no and salaries.salary > 115000 and dept_emp.to_date = "9999-01-01"
group by departments.dept_name,titles.title;
```

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

dept_name	title	total_employees
Customer Service	Engineer	10
Customer Service	Senior Engineer	15
Customer Service	Senior Staff	425
Customer Service	Staff	338
Development	Assistant Engineer	43
Development	Engineer	230
Development	Senior Engineer	247
Development	Senior Staff	2
Development	Technique Leader	13
Finance	Senior Staff	688
Finance	Staff	592
Human Resources	Senior Staff	42
Human Resources	Staff	27

Result 68

6) Assume that the company wants to reward the most senior employees who are more than 50 years of age and have contributed to the company for more than 10 years. Who is on the list? Display employee name, age, years of service in the company and joined date.

```
select first_name,last_name,TIMESTAMPDIFF(YEAR,birth_date,CURDATE())  AS age,
hire_date,TIMESTAMPDIFF(YEAR,hire_date,CURDATE()) AS years_of_service
from employees  where TIMESTAMPDIFF(YEAR,birth_date,CURDATE())>50 and
 TIMESTAMPDIFF(YEAR,hire_date,CURDATE()) >10 ;
```

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

first_name	last_name	age	hire_date	years_of_service
Georgi	Facello	68	1986-06-26	35
Bezalel	Simmel	57	1985-11-21	36
Parto	Bamford	62	1986-08-28	35
Chirstian	Koblick	67	1986-12-01	35
Kyoichi	Maliniak	67	1989-09-12	32
Anneke	Preusig	68	1989-06-02	32
Tzvetan	Zielinski	64	1989-02-10	32
Saniya	Kalloufi	63	1994-09-15	27
Sumant	Peac	69	1985-02-18	36
Duangkaew	Piveteau	58	1989-08-24	32
Mary	Sluis	68	1990-01-22	32
Patricio	Bridgland	61	1992-12-18	29
Eberhardt	Terkki	58	1985-10-20	36

Result 45

7) Find all the names (first name + last name) of employees in the database who do not work in the Human Resources department. Assume that all the people work for exactly one department.

```
select first_name,last_name
from employees,dept_emp,departments
where employees.emp_no = dept_emp.emp_no and departments.dept_no = dept_emp.dept_no and
dept_name!='Human Resources'
group by first_name,last_name;
```

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

first_name	last_name
Aamer	Anger
Aamer	Armand
Aamer	Azevdeo
Aamer	Azuma
Aamer	Baak
Aamer	Baaleh
Aamer	Baar
Aamer	Baba
Aamer	Bahl
Aamer	Bahr
Aamer	Basawa
Aamer	Bashian

Result 23

8) Find the names of all employees in the database who earn more than every employee in the Finance department. Assume that all people work for at most one company.

```
select distinct first_name,last_name
from employees,salaries,departments,dept_emp
where employees.emp_no = dept_emp.emp_no and employees.emp_no = salaries.emp_no and
departments.dept_no = dept_emp.dept_no and
salary > (select max(salary) from employees,salaries,departments,dept_emp
 where employees.emp_no = dept_emp.emp_no and employees.emp_no = salaries.emp_no and
departments.dept_no = dept_emp.dept_no and dept_name = 'Finance');
```

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

first_name	last_name
Charmane	Griswold
Boalin	Rosen
Nikolaus	Businaro
JoAnne	Matheson
Wonhee	Pagter
Tadanori	Sudbeck
Weicheng	Hatcliff
Chaitali	Baik
Mitsuyuki	Stanfel
Dines	Giaccio
Arnd	Junot

Result 53

9) Find the names of all employees who earn more than the average salary of all employees of their company.

e17153queries*

Limit to 1000 rows

```
select distinct first_name,last_name
from employees,salaries
where employees.emp_no = salaries.emp_no and
salary > (select avg(salary) from salaries );
```

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

first_name	last_name
Krassimir	Linares
Wonhee	Perl
Nidapan	Provine
Margareta	Petersohn
Urs	Krone
Franziska	Marrevee
Eishiro	Garigliano
Mary	Gente
Chinhyun	Hiyoshi
Shmuel	Sudkamp
Vivian	Chachaty
Zengping	Poehlman
Toshiki	Szilard
Matt	Benner

Result 56

10) Compute the difference between the average salary of a Senior Engineer and the average salary of all employees (including Senior Engineers).

```
select
(select avg(salary) from employees,salaries,titles where titles.emp_no = salaries.emp_no and
employees.emp_no = titles.emp_no and titles.title= 'Senior Engineer') -
(select avg(salary) from salaries)as difference;
```

difference
-3297.7505

11) Create a view current dept emp (emp no, fromdate, todate) to show only the current department for each employee. You may have to use two views for this

```
create view current_dept_emp as
select dept_emp.emp_no, dept_emp.from_date, dept_emp.to_date
from dept_emp
where dept_emp.to_date = '9999-01-01';
select * from current_dept_emp;
```

emp_no	from_date	to_date
10001	1986-06-26	9999-01-01
10002	1996-08-03	9999-01-01
10003	1995-12-03	9999-01-01
10004	1986-12-01	9999-01-01
10005	1989-09-12	9999-01-01
10006	1990-08-05	9999-01-01
10007	1989-02-10	9999-01-01
10009	1985-02-18	9999-01-01
10010	2000-06-26	9999-01-01
10012	1992-12-18	9999-01-01
10013	1985-10-20	9999-01-01
10014	1993-12-29	9999-01-01
10016	1998-02-11	9999-01-01
10017	1993-08-03	9999-01-01
10018	1992-07-29	9999-01-01

12) Write a normal SQL query to do the above task in problem 11.

e17153queries*

Limit to 1000 rows

```
SELECT dept_emp.emp_no, dept_emp.from_date, dept_emp.to_date
FROM dept_emp
WHERE dept_emp.to_date = '9999-01-01';
```

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

emp_no	from_date	to_date
10001	1986-06-26	9999-01-01
10002	1996-08-03	9999-01-01
10003	1995-12-03	9999-01-01
10004	1986-12-01	9999-01-01
10005	1989-09-12	9999-01-01
10006	1990-08-05	9999-01-01
10007	1989-02-10	9999-01-01
10009	1985-02-18	9999-01-01
10010	2000-06-26	9999-01-01
10012	1992-12-18	9999-01-01
10013	1985-10-20	9999-01-01
10014	1993-12-29	9999-01-01
10016	1998-02-11	9999-01-01
10017	1993-08-03	9999-01-01
10018	1992-07-29	9999-01-01
10019	1999-04-30	9999-01-01
10020	1997-12-30	9999-01-01

dept_emp 13

13) Create a trigger to print salary changes of employees. For example, if you enter an SQL statement such as UPDATE salaries SET salary = salary + 1000 WHERE emp no = 1500, the trigger should fire once for each row that is updated and it should print the new and old salaries, and the difference.

```
CREATE TABLE salary_change(
sc_emp_no int,
old_salary int,
new_salary int,
salary_diff int,
PRIMARY KEY (sc_emp_no),
FOREIGN KEY (sc_emp_no) REFERENCES employees(emp_no) ON DELETE CASCADE
);
DELIMITER $$
CREATE OR REPLACE TRIGGER print_salary
AFTER UPDATE ON salaries
FOR EACH ROW
BEGIN
IF New.salaries.salary <> old.salaries.salary THEN
insert into salary_change(sc_emp_no,old_salary,new_salary,salary_diff) Values
(salaries.emp_no,:old..salary,new.salary,((new.salary)-(old.salary)));
END IF
END;
DELIMITER;
SELECT *
FROM salary_changes
```

14) Create a trigger that will cause an error when an update occurs that would result in a salary increase greater than 10% of the current salary.

```
DELIMITER $$
CREATE TRIGGER salary_increase BEFORE UPDATE ON salaries FOR EACH ROW BEGIN
IF NEW.salary - OLD.salary > 0.1 * OLD.salary THEN
SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Salary increment is greater than 10% of current salary!';
END IF;
END$$
DELIMITER ;
```