Name: Ayush patel  Enrolment No. 22162171038  Class B batch 35

**Institute of Computer Technology**
**Ganpat University**
**(2CSE301) DATABASE MANAGEMENT SYSTEM**

## Practical 8 MySQL Views (2)

**Database and tables:**
https://drive.google.com/file/d/12ukAhPUqcms9dobK_XKLQgamMO9vVCn0/view?usp=sharing

1. Create a view that contains the addresses (department_name, location_id, street_address, city, state_province, country_name) of all the departments

```
CREATE VIEW department_addresses AS
SELECT
  d.DEPARTMENT_NAME,
  l.LOCATION_ID,
  l.STREET_ADDRESS,
  l.CITY,
  l.STATE_PROVINCE,
  c.COUNTRY_NAME
FROM
  departments d
  JOIN locations l ON d.LOCATION_ID = l.LOCATION_ID
  JOIN countries c ON l.COUNTRY_ID = c.COUNTRY_ID;

select * from department_addresses;
```

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

| DEPARTMENT_NAME | LOCATION_ID | STREET_ADDRESS | CITY | STATE_PROVINCE | COUNTRY_NAME |
|---|---|---|---|---|---|
| Administration | 1700 | 2004 Charade Rd | Seattle | Washington | United States of America |
| Marketing | 1800 | 147 Spadina Ave | Toronto | Ontario | Canada |
| Purchasing | 1700 | 2004 Charade Rd | Seattle | Washington | United States of America |
| Human Resources | 2400 | 8204 Arthur St | London | | United Kingdom |
| Shipping | 1500 | 2011 Interiors Blvd | South San Francisco | California | United States of America |
| IT | 1400 | 2014 Jabberwocky Rd | Southlake | Texas | United States of America |
| Public Relations | 2700 | Schwanthalerstr. 7031 | Munich | Bavaria | Germany |
| Executive | 1700 | 2004 Charade Rd | Seattle | Washington | United States of America |

department_addresses 10 ×

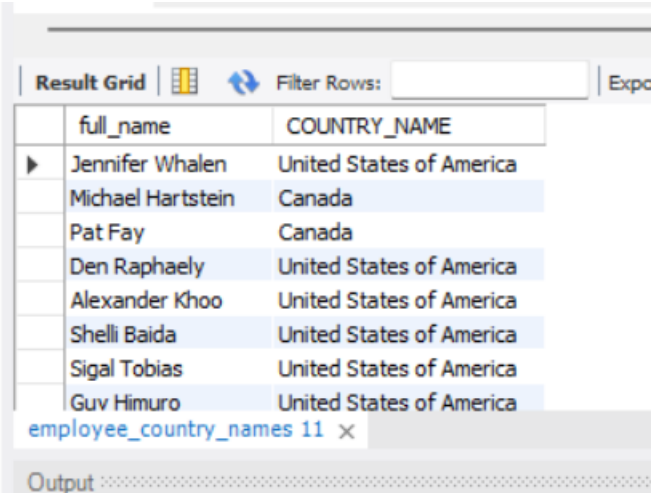
Output

2. Create a view that contains the full name (e.g. Steven King) of all employees along with the country in which they are working

```
CREATE VIEW employee_country_names AS
SELECT
  CONCAT(e.FIRST_NAME, ' ', e.LAST_NAME) AS full_name,
```

```
   c.COUNTRY_NAME
FROM
   employees e
   JOIN departments d ON e.DEPARTMENT_ID= d.DEPARTMENT_ID
   JOIN locations l ON d.LOCATION_ID = l.LOCATION_ID
   JOIN countries c ON l.COUNTRY_ID = c.COUNTRY_ID;

select * from employee_country_names;
```




**3.** Create a view to find the name (first_name, last_name) and salary of the employees whose salary is greater than the average salary.

```
CREATE VIEW high_salary_employees AS
SELECT
   FIRST_NAME,
  LAST_NAME,
   SALARY
FROM
   employees
WHERE
 SALARY > (SELECT AVG(SALARY) FROM employees);

select * from high_salary_employees;
```

Name: Ayush patel  Enrolment No. 22162171038  Class B batch 35

```
43 •  select * from high_salary_employees;
```

Result Grid | Filter Rows: | Export:

| FIRST_NAME | LAST_NAME | SALARY |
|---|---|---|
| Steven | King | 24000.00 |
| Neena | Kochhar | 17000.00 |
| Lex | De Haan | 17000.00 |
| Alexander | Hunold | 9000.00 |
| Nancy | Greenberg | 12000.00 |
| Daniel | Faviet | 9000.00 |
| John | Chen | 8200.00 |
| Ismael | Sciarra | 7700.00 |

high_salary_employees 12 ×

4. Create a view to get the job_ID, JOB_title and maximum salary of the employees where the maximum salary is greater than or equal to $4000.

```
CREATE VIEW high_salary_jobs AS
SELECT
  j.JOB_ID,
  j.JOB_TITLE,
  MAX(e.SALARY) AS max_SALARY
FROM
  employees e
  JOIN jobs j ON e.JOB_ID = j.JOB_ID
GROUP BY
  j.JOB_ID, j.JOB_TITLE
HAVING
  MAX(e.SALARY) >= 4000;

select * from high_salary_jobs;
```

Result Grid | Filter Rows: | Export:

| JOB_ID | JOB_TITLE | max_SALARY |
|---|---|---|
| AD_PRES | President | 24000.00 |
| AD_VP | Administration Vice President | 17000.00 |
| IT_PROG | Programmer | 9000.00 |
| FI_MGR | Finance Manager | 12000.00 |
| FI_ACCOUNT | Accountant | 9000.00 |
| PU_MAN | Purchasing Manager | 11000.00 |
| ST_MAN | Stock Manager | 8200.00 |
| SA_MAN | Sales Manager | 14000.00 |

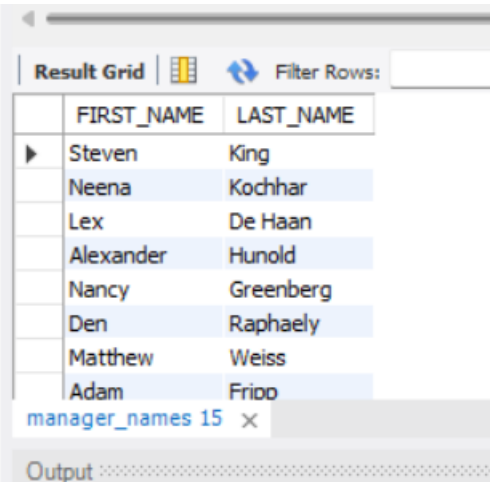
high_salary_jobs 14 ×

Output

5. Create a view that displays the first_name and last_name of employees who are managers.

Name: Ayush patel  Enrolment No. 22162171038  Class B batch 35

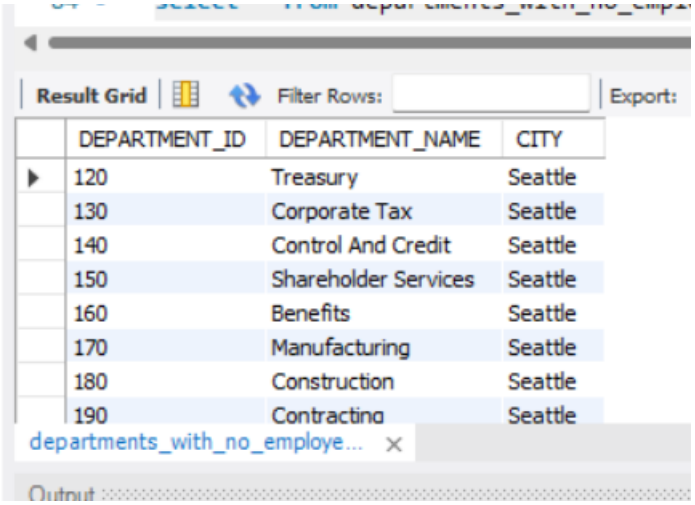
```
CREATE VIEW manager_names AS
SELECT
   e.FIRST_NAME,
   e.LAST_NAME
FROM
   employees e
WHERE
   e.EMPLOYEE_ID IN (SELECT DISTINCT MANAGER_ID FROM employees);
select * from manager_names ;
```




**6.** Create a view to list the department ID, department_name and city of all the departments where no employee is working.

```
CREATE VIEW departments_with_no_employees AS
SELECT
   d.DEPARTMENT_ID,
   d.DEPARTMENT_NAME,
   l.CITY
FROM
   departments d
   JOIN locations l ON d.LOCATION_ID = l.LOCATION_ID
LEFT JOIN
   employees e ON d.DEPARTMENT_ID = e.DEPARTMENT_ID
WHERE
   e.EMPLOYEE_ID IS NULL;
select * from departments_with_no_employees;
```

Name: Ayush patel  Enrolment No. 22162171038  Class B batch 35



Result Grid | Filter Rows: | Export:

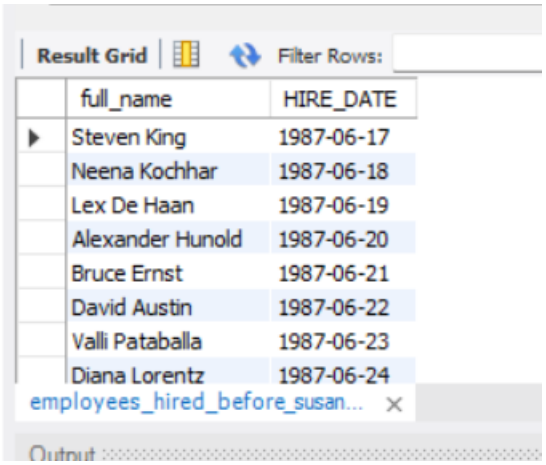
| DEPARTMENT_ID | DEPARTMENT_NAME | CITY |
|---|---|---|
| 120 | Treasury | Seattle |
| 130 | Corporate Tax | Seattle |
| 140 | Control And Credit | Seattle |
| 150 | Shareholder Services | Seattle |
| 160 | Benefits | Seattle |
| 170 | Manufacturing | Seattle |
| 180 | Construction | Seattle |
| 190 | Contracting | Seattle |

departments_with_no_employe...

Output

**7.** Create a view to list the full names and hire_date of the employees who were hired before Susan Mavris.

```
CREATE VIEW employees_hired_before_susan AS
SELECT
  CONCAT(FIRST_NAME, ' ', LAST_NAME) AS full_name,HIRE_DATE
FROM
  employees
WHERE
  HIRE_DATE < (SELECT HIRE_DATE FROM employees WHERE FIRST_NAME =
    'Susan' AND LAST_NAME = 'Mavris');
select * from employees_hired_before_susan;
```



Result Grid | Filter Rows:

| full_name | HIRE_DATE |
|---|---|
| Steven King | 1987-06-17 |
| Neena Kochhar | 1987-06-18 |
| Lex De Haan | 1987-06-19 |
| Alexander Hunold | 1987-06-20 |
| Bruce Ernst | 1987-06-21 |
| David Austin | 1987-06-22 |
| Valli Pataballa | 1987-06-23 |
| Diana Lorentz | 1987-06-24 |

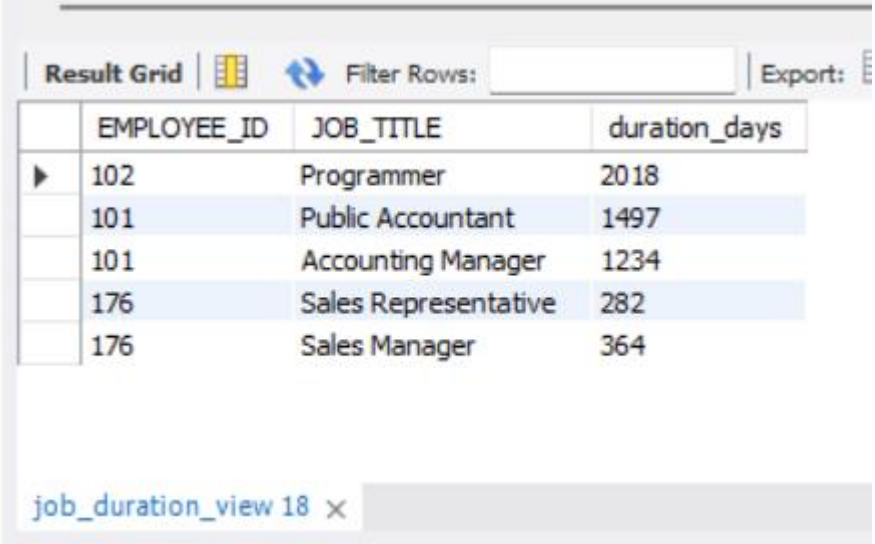
employees_hired_before_susan...

Output

**8.** Create a view to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 80 and 90.

```
CREATE VIEW job_duration_view AS
SELECT
  e.EMPLOYEE_ID,
  j.JOB_TITLE,
  DATEDIFF(jh.END_DATE, jh.START_DATE) AS duration_days
FROM
```

```
  employees e
  JOIN job_history jh ON e.EMPLOYEE_ID = jh.EMPLOYEE_ID
  JOIN jobs j ON jh.JOB_ID = j.JOB_ID
WHERE
  e.DEPARTMENT_ID IN (80, 90);
select * from job_duration_view;
```



Result Grid | Filter Rows: | Export:

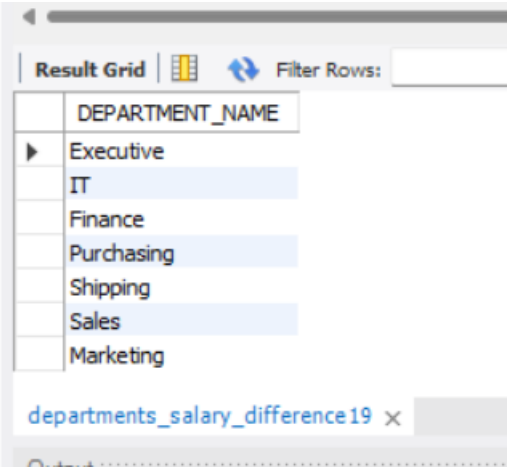
| EMPLOYEE_ID | JOB_TITLE | duration_days |
|---|---|---|
| 102 | Programmer | 2018 |
| 101 | Public Accountant | 1497 |
| 101 | Accounting Manager | 1234 |
| 176 | Sales Representative | 282 |
| 176 | Sales Manager | 364 |

job_duration_view 18

**9.** Create a view to display the department names in which the difference between minimum and maximum salary is more than 4000

```
CREATE VIEW departments_salary_difference AS
SELECT
  d.DEPARTMENT_NAME
FROM
  departments d
  JOIN employees e ON d.DEPARTMENT_ID = e.DEPARTMENT_ID
GROUP BY
  d.DEPARTMENT_NAME
HAVING
  MAX(e.SALARY) - MIN(e.SALARY) > 4000;
select * from departments_salary_difference;
```

Name: Ayush patel  Enrolment No. 22162171038  Class B batch 35




**10.**For all above created views, Update at least 1 record from each and verify whether the update gets reflected in the base table(s) or not

```
UPDATE high_salary_employees
SET SALARY = SALARY + 1000
WHERE FIRST_NAME = 'John' AND LAST_NAME = 'Doe';
```