



Data Analysis and Integration

Lab 2: SQL Views and Data Integration

Note: This lab assumes that you have previously completed Lab 1.

Creating the views

1. Open a terminal and connect to the local MySQL server: **mysql -u aid -p**
Password: **aid**
2. On the MySQL prompt, execute the following command to connect to the database:
use employees
3. Execute the following command to show the tables in the database:
show tables;
4. Take a moment to inspect the contents of these tables:
 - **select * from salaries limit 20;**
 - **select * from dept_emp limit 20;**
 - **select * from dept_manager;**
 - **select * from titles limit 20;**You will notice these tables have **from_date** and **to_date** columns to indicate the time period when those facts were applicable.

5. Execute the following query to obtain the current salary of each employee:

```
select emp_no, salary
from salaries
where from_date <= current_date and to_date >= current_date
limit 20;
```

6. Create a view that returns the current salary of each employee:

```
create or replace view curr_salaries(emp_no, salary) as
select emp_no, salary
from salaries
where from_date <= current_date and to_date >= current_date;
```

7. Execute the following query to check that the view is working:

```
select * from curr_salaries limit 20;
```

8. Execute the following query to obtain the current department of each employee:

```
select emp_no, dept_no
from dept_emp
where from_date <= current_date and to_date >= current_date
limit 20;
```

9. Create a view that returns the current department of each employee:

```
create or replace view curr_dept_emp(emp_no, dept_no) as  
  select emp_no, dept_no  
  from dept_emp  
  where from_date <= current_date and to_date >= current_date;
```

10. Execute the following query to check that the view is working:

```
select * from curr_dept_emp limit 20;
```

11. Execute the following query to obtain the current manager of each department:

```
select emp_no, dept_no  
from dept_manager  
where from_date <= current_date and to_date >= current_date;
```

12. Create a view that returns the current manager of each department:

```
create or replace view curr_dept_manager(emp_no, dept_no) as  
  select emp_no, dept_no  
  from dept_manager  
  where from_date <= current_date and to_date >= current_date;
```

13. Execute the following query to check that the view is working:

```
select * from curr_dept_manager;
```

14. Execute the following query to obtain the current title of each employee:

```
select emp_no, title  
from titles  
where from_date <= current_date and to_date >= current_date  
limit 20;
```

15. Create a view that returns the current title of each employee:

```
create or replace view curr_titles(emp_no, title) as  
  select emp_no, title  
  from titles  
  where from_date <= current_date and to_date >= current_date;
```

16. Execute the following query to check that the view is working:

```
select * from curr_titles limit 20;
```

17. Execute the following command to show the tables in the database:

```
show tables;
```

These should now include the views that you have created above.

Queries over views

18. Execute the following query to get the (current) number of employees:

```
select count(distinct emp_no)  
from curr_salaries;
```

19. Execute the following query to get the (current) number of employees by department:

```
select a.dept_no, a.dept_name, count(distinct b.emp_no) as count_emp_no  
from departments as a, curr_dept_emp as b  
where a.dept_no = b.dept_no  
group by a.dept_no, a.dept_name  
order by count_emp_no desc;
```

20. Execute the following query to show the (current) sum of salaries by department:

```
select a.dept_no, a.dept_name, sum(c.salary) as sum_salary  
from departments as a, curr_dept_emp as b, curr_salaries as c  
where a.dept_no = b.dept_no and b.emp_no = c.emp_no  
group by a.dept_no, a.dept_name  
order by sum_salary desc;
```

Creating the company database

21. Download the file **company.sql** to your Downloads folder (/home/aid/Downloads).

22. Take a moment to inspect the contents of the **company.sql** script.

- Locate the CREATE DATABASE statement.
- Locate all CREATE TABLE statements.
- Check the columns and data types for each table.
- Check the primary and foreign keys for each table.
- Locate the INSERT instructions to load data into these tables.

23. Open a terminal and navigate to the folder where the file is located.

24. Execute the following command to login to the local MySQL server: **mysql -u aid -p**
Password: **aid**

25. On the MySQL prompt, execute the following command to create the database:
source company.sql

26. Execute the following command to change to the company database:
use company

27. Execute the following command to show the tables in the database:
show tables;

28. Take a moment to inspect the contents of each table:

- **select * from employees;**
- **select * from department;**
- **select * from branches;**

Creating the mediated schema

We will create the following global views to integrate data from both databases:

all_employees(emp_no, first_name, last_name, birth_date, report_to)	Returns the list of all employees from both databases
all_departments(dept_no, dept_name)	Returns the list of all departments from both databases
all_dept_emp(emp_no, dept_no)	Returns the department for all employees in both databases
all_salaries(emp_no, salary)	Returns the salaries for all employees in both databases
all_titles(emp_no, title)	Returns the titles for all employees in both databases

29. To query tables from multiple databases, we will use the database name as a prefix. For example:

```
select * from employees.employees limit 10;
select * from company.employees limit 10;
```

30. Run the following union query to retrieve job titles from both databases:

Note: For testing purposes, we are limiting the results to 10 records from each database.

```
(select emp_no, title from employees.curr_titles limit 10)
union
(select employeeid, jobtitle from company.employees limit 10);
```

31. Create the following view to retrieve job titles from both databases:

```
create or replace view all_titles(emp_no, title) as
(select emp_no, title from employees.curr_titles)
union
(select employeeid, jobtitle from company.employees);
```

32. Test the view by running the following query:

```
select * from all_titles limit 20;
```

33. Run the following union query to retrieve salaries from both databases:

Note: For testing purposes, we are limiting the results to 10 records from each database.

```
(select emp_no, salary from employees.curr_salaries limit 10)
union
(select employeeid, salary from company.employees limit 10);
```

34. Create the following view to retrieve salaries from both databases:

```
create or replace view all_salaries(emp_no, salary) as  
  (select emp_no, salary from employees.curr_salaries)  
union  
  (select employeeid, salary from company.employees);
```

35. Test the view by running the following query:

```
select * from all_salaries limit 20;
```

36. Run the following union query to retrieve the employees and departments from both databases:
Note: For testing purposes, we are limiting the results to 10 records from each database.

```
(select emp_no, dept_no from employees.curr_dept_emp limit 10)  
union  
(select employeeid, departmentid from company.employees limit 10);
```

37. Create the following view to retrieve the employees and departments from both databases:

```
create or replace view all_dept_emp(emp_no, dept_no) as  
  (select emp_no, dept_no from employees.curr_dept_emp)  
union  
  (select employeeid, departmentid from company.employees);
```

38. Test the view by running the following query:

```
select * from all_dept_emp limit 20;
```

39. Run the following union query to retrieve the departments from both databases:

```
(select dept_no, dept_name from employees.departments)  
union  
(select departmentid, departmentname from company.department);
```

40. Create the following view to retrieve the departments from both databases:

```
create or replace view all_departments(dept_no, dept_name) as  
  (select dept_no, dept_name from employees.departments)  
union  
  (select departmentid, departmentname from company.department);
```

41. Test the view by running the following query:

```
select * from all_departments;
```

42. Run the following union query to retrieve the employees from both databases:
Note: For testing purposes, we are limiting the results to 10 records from each database.

```
(select a.emp_no, a.first_name, a.last_name, a.birth_date, c.emp_no as reportto  
from employees.employees as a,  
  employees.curr_dept_emp as b,  
  employees.curr_dept_manager as c
```

```

where a.emp_no = b.emp_no and b.dept_no = c.dept_no
limit 10)
union
(select employeeid, firstname, lastname, dob, reportto
from company.employees
limit 10);

```

43. Create the following view to retrieve the employees from both databases:

```

create or replace view all_employees(emp_no, first_name, last_name, birth_date, report_to) as
(select a.emp_no, a.first_name, a.last_name, a.birth_date, c.emp_no as reportto
from employees.employees as a,
      employees.curr_dept_emp as b,
      employees.curr_dept_manager as c
where a.emp_no = b.emp_no and b.dept_no = c.dept_no)
union
(select employeeid, firstname, lastname, dob, reportto
from company.employees);

```

44. Test the view by running the following query:

```

select * from all_employees limit 20;

```

45. Execute the following command to show the tables in the database:

```

show tables;

```

These should now include the views that you have created above.

Queries over the mediated schema

46. Execute the following query to get the (current) total number of employees:

```

select count(distinct emp_no) as count_emp_no
from all_salaries;

```

47. Execute the following query to get the (current) number of employees by department:

```

select a.dept_no, a.dept_name, count(distinct b.emp_no) as count_emp_no
from all_departments as a, all_dept_emp as b
where a.dept_no = b.dept_no
group by a.dept_no, a.dept_name;

```

48. Execute the following query to show the (current) sum of salaries by department:

```

select a.dept_no, a.dept_name, sum(c.salary) as sum_salary
from all_departments as a, all_dept_emp as b, all_salaries as c
where a.dept_no = b.dept_no and b.emp_no = c.emp_no
group by a.dept_no, a.dept_name;

```

49. Compare the results of these queries with the ones that you have performed earlier over the **employees** database alone. Check the differences in terms of employees, departments, and salaries.

Exercise


50. Get the list of department names:
- from the **employees** database (using the **departments** table)
 - from the **company** database (using the **department** table)
 - from both databases (using the **all_departments** view)
- Sort the results alphabetically.

```
+-----+
| dept_name |
+-----+
| Customer Service |
| Development |
| Finance |
| Human Resources |
| Marketing |
| Production |
| Quality Management |
| Research |
| Sales |
+-----+
9 rows in set (0.00 sec)
```

```
+-----+
| departmentname |
+-----+
| Finance |
| HR |
| IT |
| marketing |
| Sales |
+-----+
5 rows in set (0.00 sec)
```

```
+-----+
| dept_name |
+-----+
| Customer Service |
| Development |
| Finance |
| Finance |
| HR |
| Human Resources |
| IT |
| marketing |
| Marketing |
| Production |
| Quality Management |
| Research |
| Sales |
| Sales |
+-----+
14 rows in set (0.00 sec)
```

51. Compare the results and identify similar department names.

52. Get the list of (distinct) job titles:
- from the **employees** database (using the **titles** table),
 - from the **company** database (using the **employees** table),
-  → c) from both databases (using the **all_titles** view) and take a screenshot.
- Sort the results alphabetically.

```
+-----+
| title |
+-----+
| Assistant Engineer |
| Engineer |
| Manager |
| Senior Engineer |
| Senior Staff |
| Staff |
| Technique Leader |
+-----+
7 rows in set (0.00 sec)
```

```
+-----+
| jobtitle |
+-----+
| Accountant |
| Admin |
| CEO |
| Director |
| Network Engineer |
| President |
| Reporting Manager |
| Sales Manager |
| Sales Rep |
| Software Engineer |
| Sr. Manager |
| Team Leader |
| Vice President |
+-----+
13 rows in set (0.00 sec)
```

```
+-----+
| title |
+-----+
| Accountant |
| Admin |
| Assistant Engineer |
| CEO |
| Director |
| Engineer |
| Manager |
| Network Engineer |
| President |
| Reporting Manager |
| Sales Manager |
| Sales Rep |
| Senior Engineer |
| Senior Staff |
| Software Engineer |
| Sr. Manager |
| Staff |
| Team Leader |
| Technique Leader |
| Vice President |
+-----+
20 rows in set (0.00 sec)
```

53. Compare the results and identify similar job titles.