

# Data Analysis and Integration

---

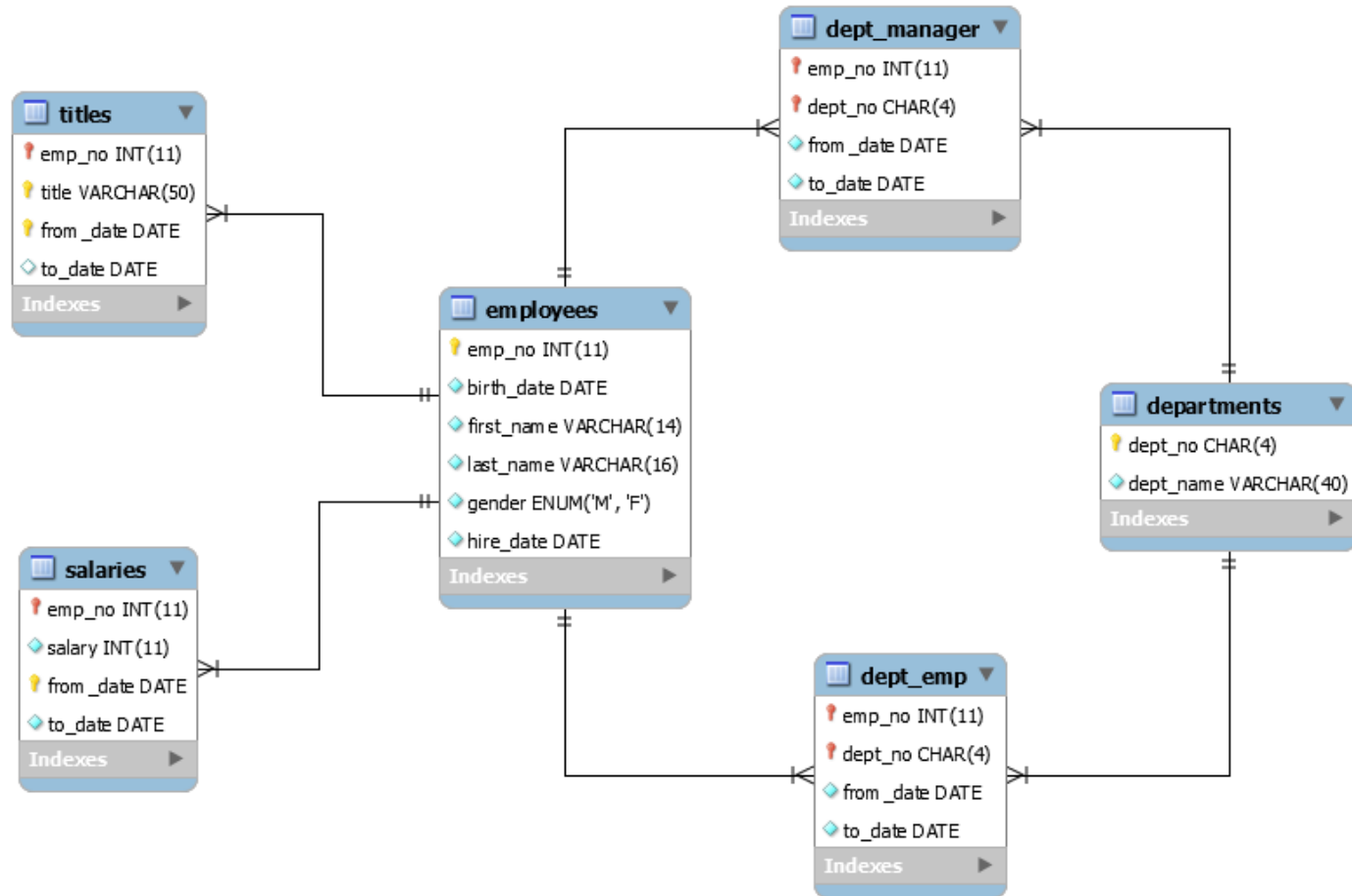
Concepts of data integration

# Introduction

- The need for data integration
  - company A merges with company B
  - A is a company with the **employees** database
  - B is a company with the **company** database
  - provide an integrated view of data from both companies
    - e.g. employees, departments, salaries, job titles

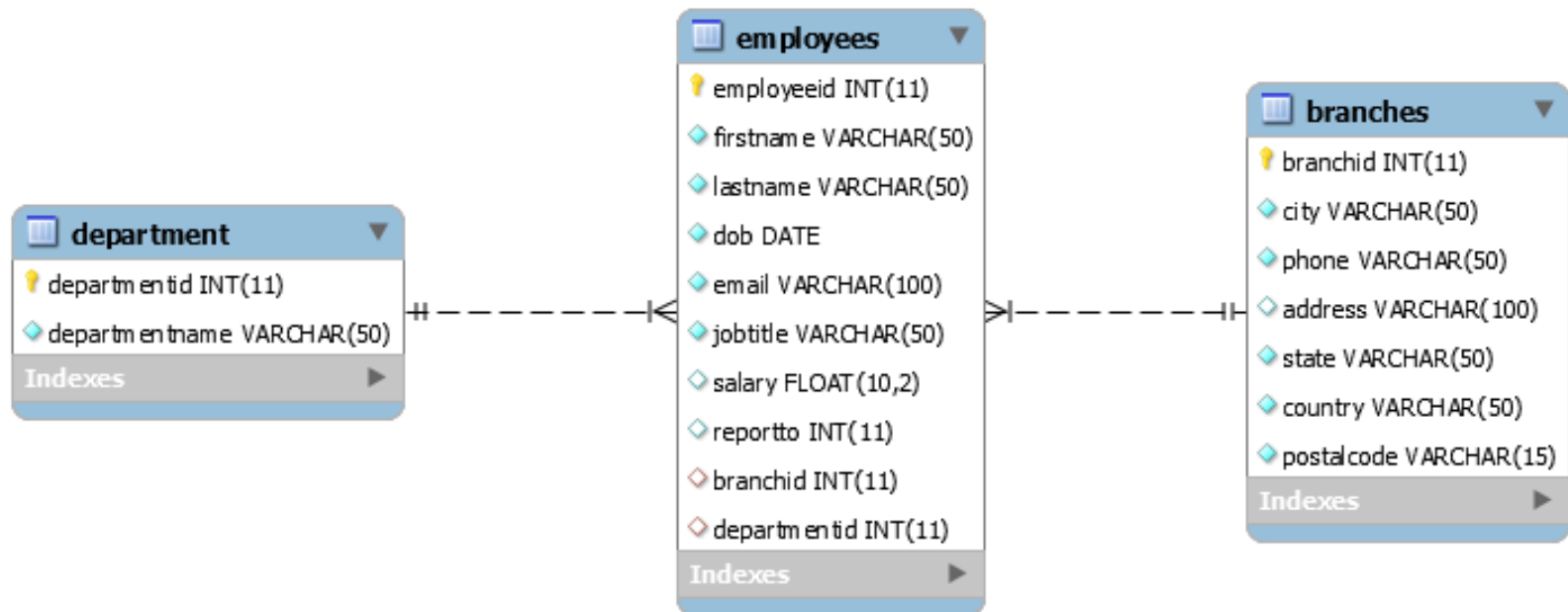
# Company A

- The employees database



# Company B

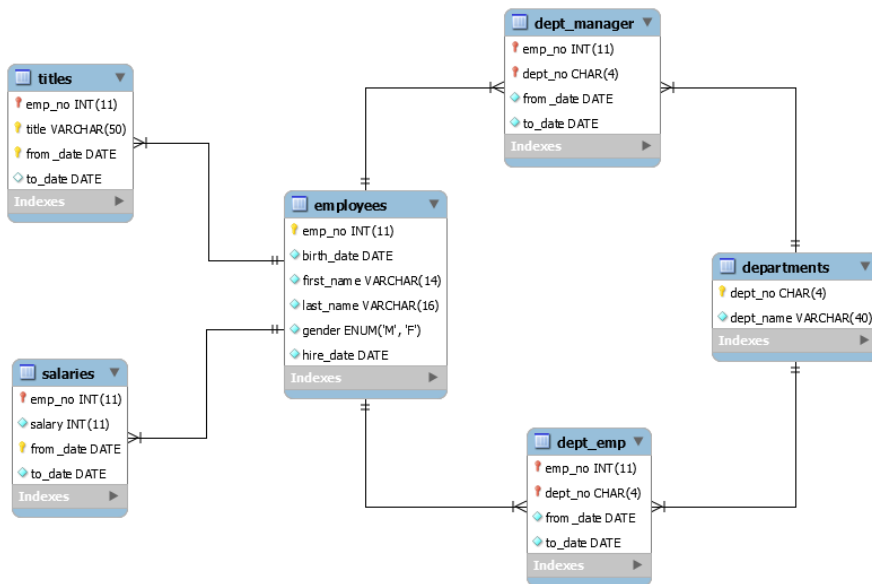
- The company database



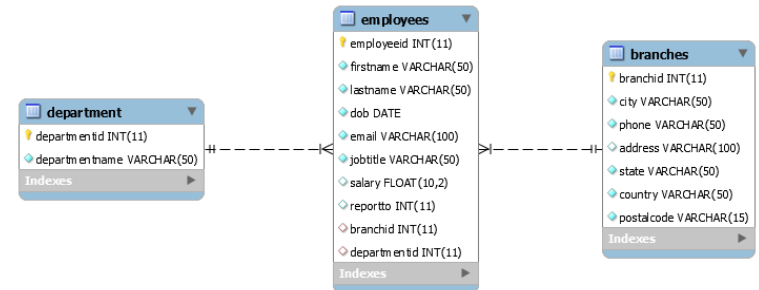
# Data sources

- Comparing the two **data sources** (schema)

data source A (**employees** database)

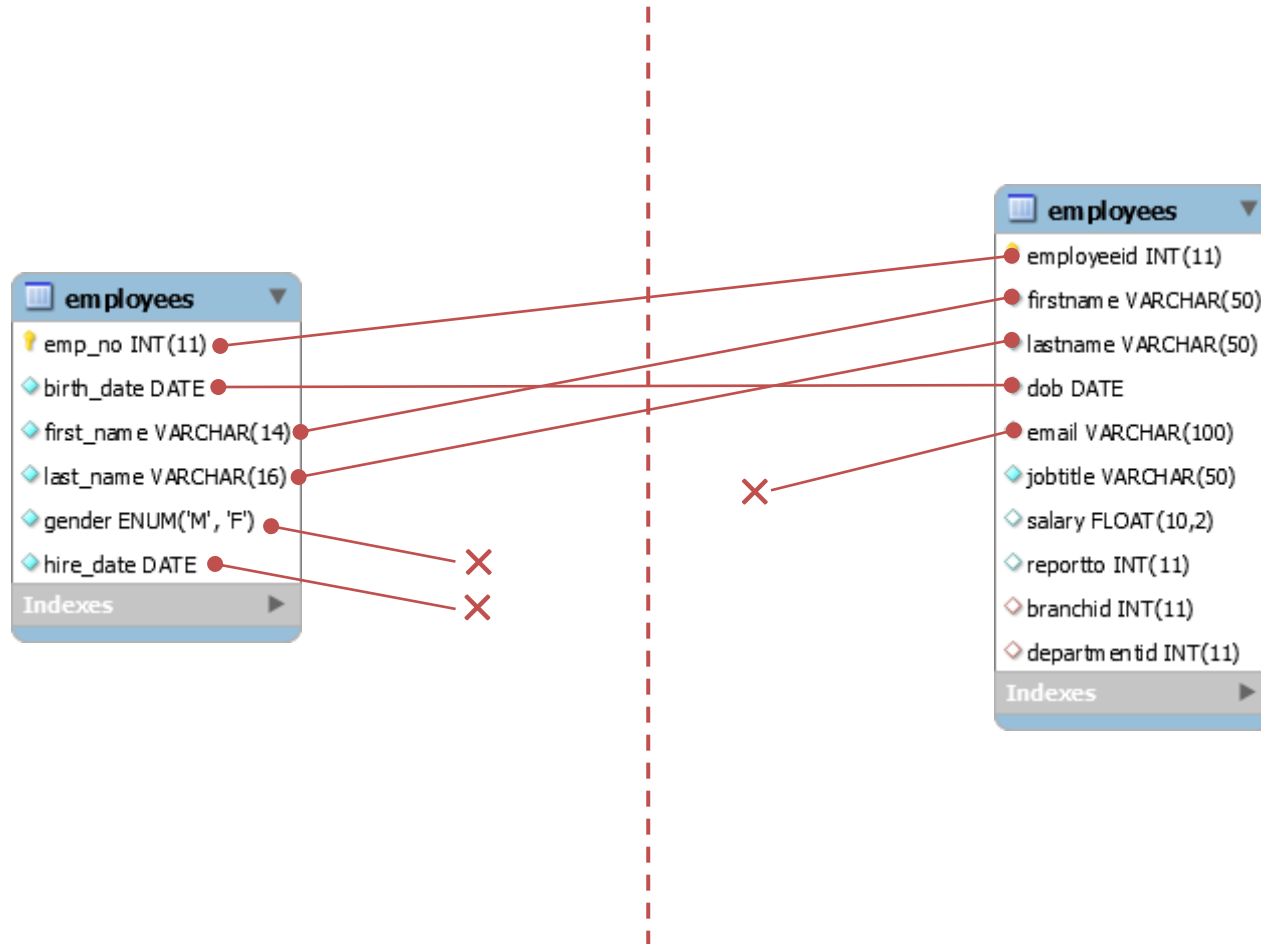


data source B (**company** database)



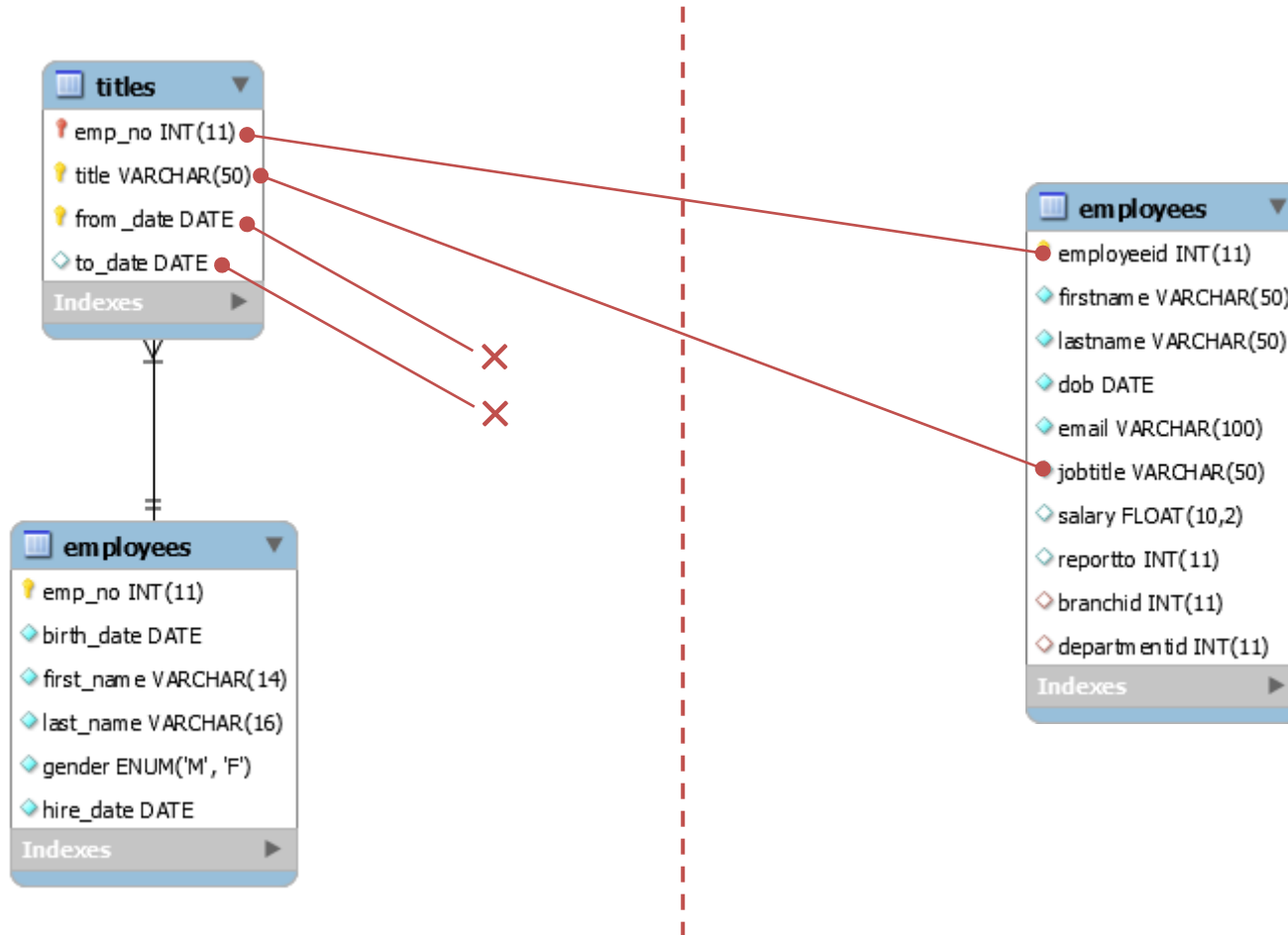
# Schema matching (employees)

- Comparing the two data sources (**schema matching**)



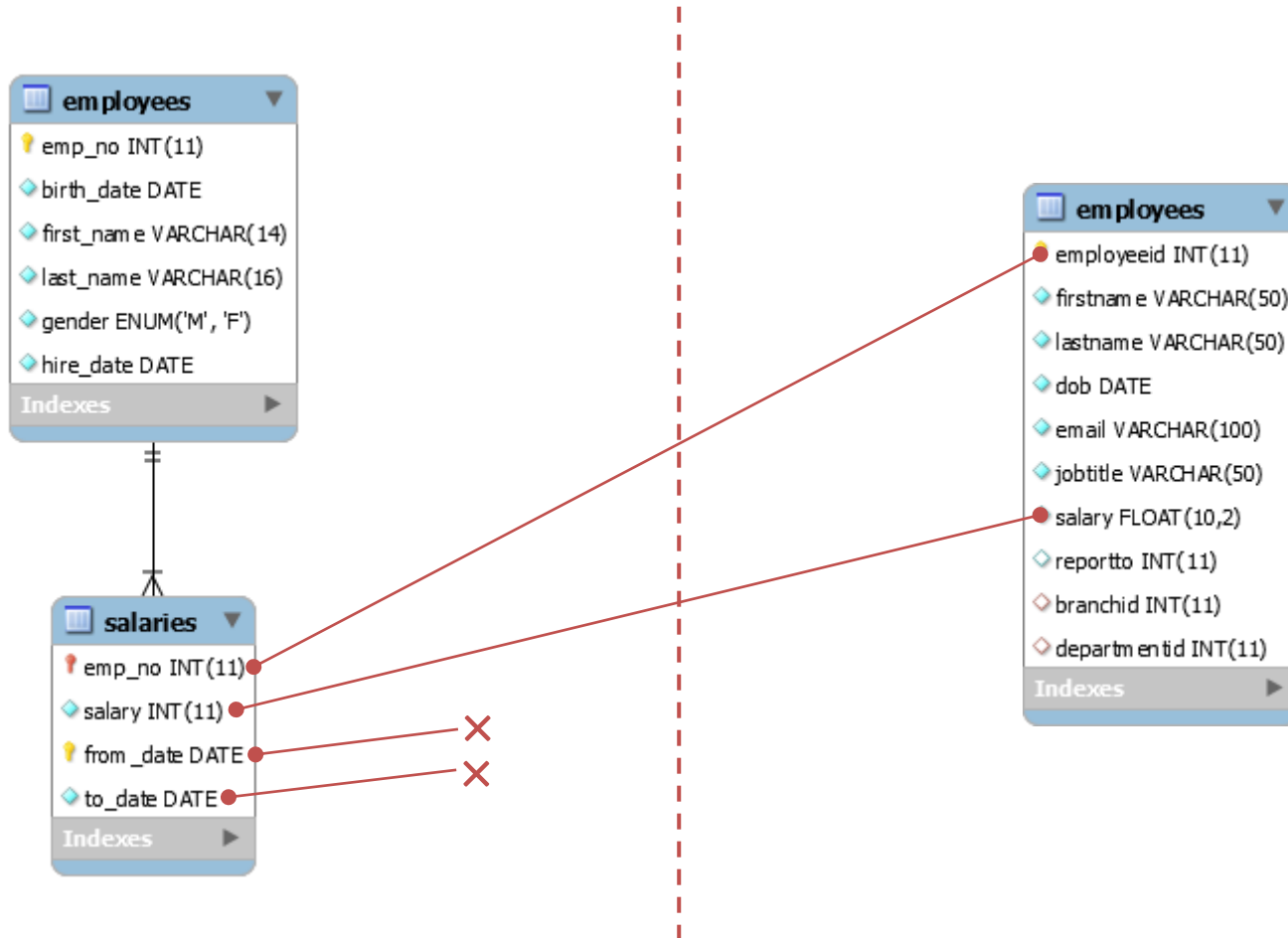
# Schema matching (titles)

- Comparing the two data sources (**schema matching**)



# Schema matching (salaries)

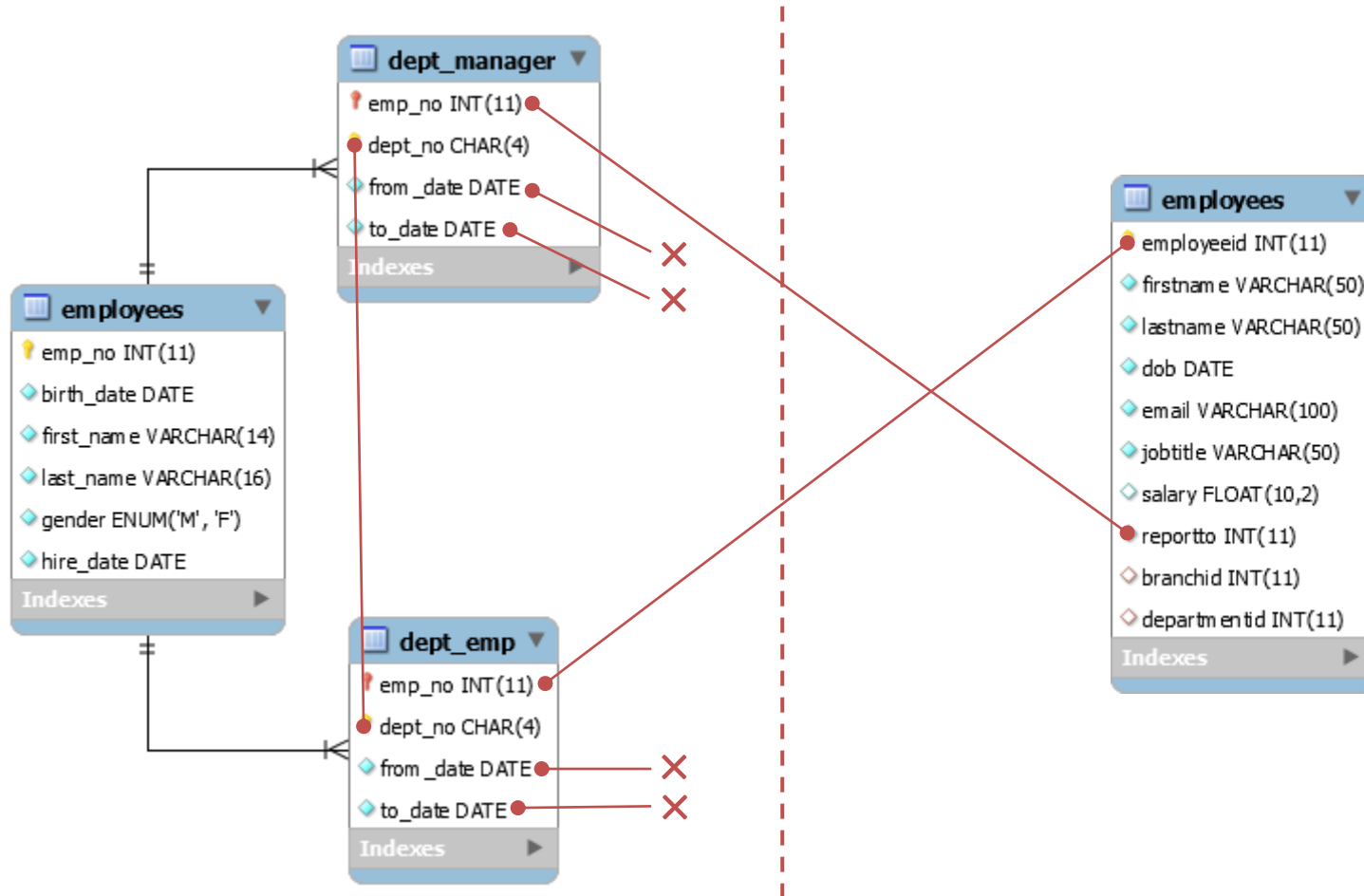
- Comparing the two data sources (**schema matching**)





# Schema matching (managers)

- Comparing the two data sources (**schema matching**)



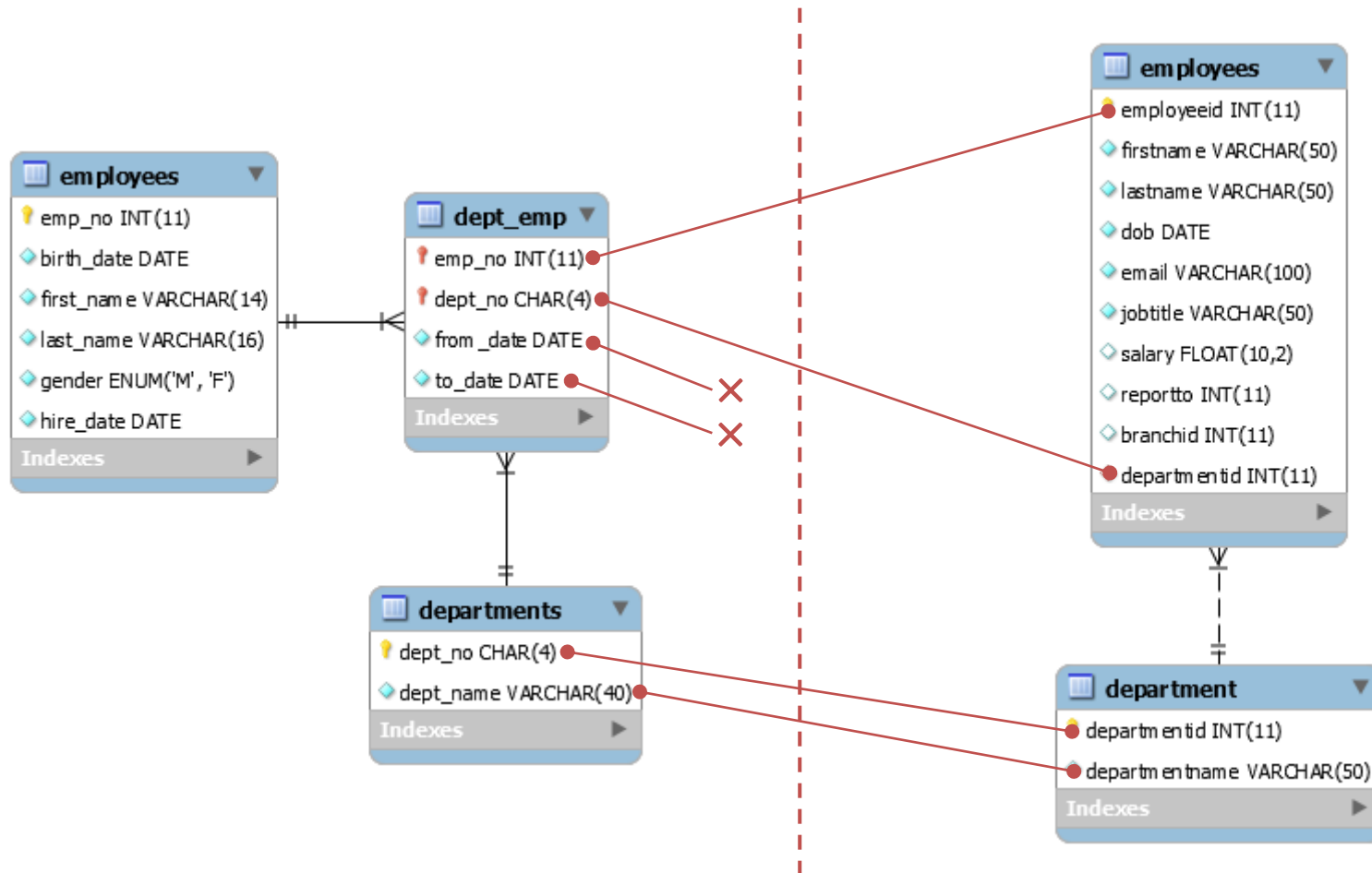
# Schema matching (branches)

- Comparing the two data sources (**schema matching**)



# Schema matching (departments)

- Comparing the two data sources (**schema matching**)



# Mediated schema

- Once company A merges with company B
  - we need to access data through a single/uniform access point
- Example of a common schema (**mediated schema**)
  - all\_employees(emp\_no, first\_name, last\_name, birth\_date, report\_to)
  - all\_departments(dept\_no, dept\_name)
  - all\_dept\_emp(emp\_no, dept\_no)
  - all\_salaries(emp\_no, salary)
  - all\_titles(emp\_no, title)

# Mediated schema

- Common schema (**mediated schema**)

all\_employees(emp\_no, first\_name, last\_name, birth\_date, report\_to)

all\_departments(dept\_no, dept\_name)

all\_dept\_emp(emp\_no, dept\_no)

all\_salaries(emp\_no, salary)

all\_titles(emp\_no, title)

- absence of **from\_date** and **to\_date** attributes means that data retrieved from the **employees** database will always refer to current date

# Wrappers for data sources

- We have a set of views to retrieve data for the current date

employees(emp\_no, birth\_date, first\_name, last\_name, gender, hire\_date)

departments(dept\_no, dept\_name)

curr\_dept\_emp(emp\_no, dept\_no)

curr\_dept\_manager(emp\_no, dept\_no)

curr\_salaries(emp\_no, salary)

curr\_titles(emp\_no, title)

- we will use these views as a **wrapper** for the employees database
  - i.e. a layer through which we access the employees database

# Schema mapping (all\_employees)

- Mapping to common schema (**schema mapping**)
  - transformations/queries that populate common schema  
all\_employees(emp\_no, first\_name, last\_name, birth\_date, report\_to)
  - from **employees** database

```
select a.emp_no, a.first_name, a.last_name, a.birth_date, c.emp_no
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;
```
  - from **company** database

```
select employeeid, firstname, lastname, dob, reportto
from company.employees;
```

# Schema mapping (all\_employees)

- Mapping to common schema (**schema mapping**)

all\_employees(emp\_no, first\_name, last\_name, birth\_date, report\_to)

```
(select a.emp_no, a.first_name, a.last_name, a.birth_date, c.emp_no
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);
```

| emp_no | first_name | last_name | birth_date | emp_no |
|--------|------------|-----------|------------|--------|
| 21637  | Yefim      | Luby      | 1964-04-28 | 110039 |
| 25949  | Owen       | Matheson  | 1959-08-08 | 110039 |
| ...    | ...        | ...       | ...        | ...    |
| 1001   | Ravi       | Gupta     | 1969-12-03 | 1001   |
| 1002   | Ram        | charan    | 1985-02-20 | 1001   |
| ...    | ...        | ...       | ...        | ...    |



# Schema mapping (all\_employees)

- Mapping to common schema (**schema mapping**)

all\_employees(emp\_no, first\_name, last\_name, birth\_date, report\_to)

```
create 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
   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);
```

# Schema mapping (all\_departments)

- Mapping to common schema (**schema mapping**)

all\_departments(dept\_no, dept\_name)

– from **employees** database

```
select dept_no, dept_name  
from employees.departments
```

– from **company** database

```
select departmentid, departmentname  
from company.department
```

# Schema mapping (all\_departments)

- Mapping to common schema (**schema mapping**)

all\_departments(dept\_no, dept\_name)

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

| dept_no | dept_name          |
|---------|--------------------|
| d009    | Customer Service   |
| d005    | Development        |
| d002    | Finance            |
| d003    | Human Resources    |
| d001    | Marketing          |
| d004    | Production         |
| d006    | Quality Management |
| d008    | Research           |
| d007    | Sales              |
| 101     | IT                 |
| 102     | HR                 |
| 103     | Finance            |
| 104     | Sales              |
| 105     | marketing          |

14 rows in set (0.00 sec)

# Schema mapping (all\_departments)

- Mapping to common schema (**schema mapping**)

all\_departments(dept\_no, dept\_name)

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

# Schema mapping (all\_dept\_emp)

- Mapping to common schema (**schema mapping**)

all\_dept\_emp(emp\_no, dept\_no)

– from **employees** database

```
select emp_no, dept_no  
from employees.curr_dept_emp
```

– from **company** database

```
select employeeid, departmentid  
from company.employees
```

# Schema mapping (all\_dept\_emp)

- Mapping to common schema (**schema mapping**)

all\_dept\_emp(emp\_no, dept\_no)

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

| emp_no | dept_no |
|--------|---------|
| 10721  | d009    |
| 11260  | d009    |
| ...    | ...     |
| 1008   | 101     |
| 1014   | 101     |
| ...    | ...     |

# Schema mapping (all\_dept\_emp)

- Mapping to common schema (**schema mapping**)

all\_dept\_emp(emp\_no, dept\_no)

```
create 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);
```

# Schema mapping (all\_salaries)

- Mapping to common schema (**schema mapping**)

all\_salaries(emp\_no, salary)

– from **employees** database

```
select emp_no, salary  
from employees.curr_salaries
```

– from **company** database

```
select employeeid, salary  
from company.employees
```



# Schema mapping (all\_salaries)

- Mapping to common schema (**schema mapping**)  
all\_salaries(emp\_no, salary)

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

| emp_no | salary    |
|--------|-----------|
| 10721  | 44812.00  |
| 11260  | 52435.00  |
| ...    | ...       |
| 1001   | 850000.00 |
| 1002   | 650000.00 |
| ...    | ...       |

# Schema mapping (all\_salaries)

- Mapping to common schema (**schema mapping**)  
all\_salaries(emp\_no, salary)

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

# Schema mapping (all\_titles)

- Mapping to common schema (**schema mapping**)

all\_titles(emp\_no, title)

– from **employees** database

```
select emp_no, title  
from employees.curr_titles
```

– from **company** database

```
select employeeid, jobtitle  
from company.employees
```

# Schema mapping (all\_titles)

- Mapping to common schema (**schema mapping**)

all\_titles(emp\_no, title)

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

| emp_no | title              |
|--------|--------------------|
| 11371  | Senior Engineer    |
| 41548  | Staff              |
| 62635  | Engineer           |
| 64387  | Senior Staff       |
| 110039 | Manager            |
| 204631 | Assistant Engineer |
| 207968 | Technique Leader   |
| ...    | ...                |
| 1001   | CEO                |
| 1002   | Director           |
| 1003   | President          |
| 1004   | Vice President     |
| 1005   | Sr. Manager        |
| 1007   | Sales Manager      |
| 1008   | Reporting Manager  |
| 1009   | Team Leader        |
| 1010   | Sales Rep          |
| 1014   | Software Engineer  |
| 1023   | Admin              |
| 1024   | Network Engineer   |
| ...    | ...                |

# Schema mapping (all\_titles)

- Mapping to common schema (**schema mapping**)

all\_titles(emp\_no, title)

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

# Data matching – duplicates

- When comparing data instances from distinct data sources

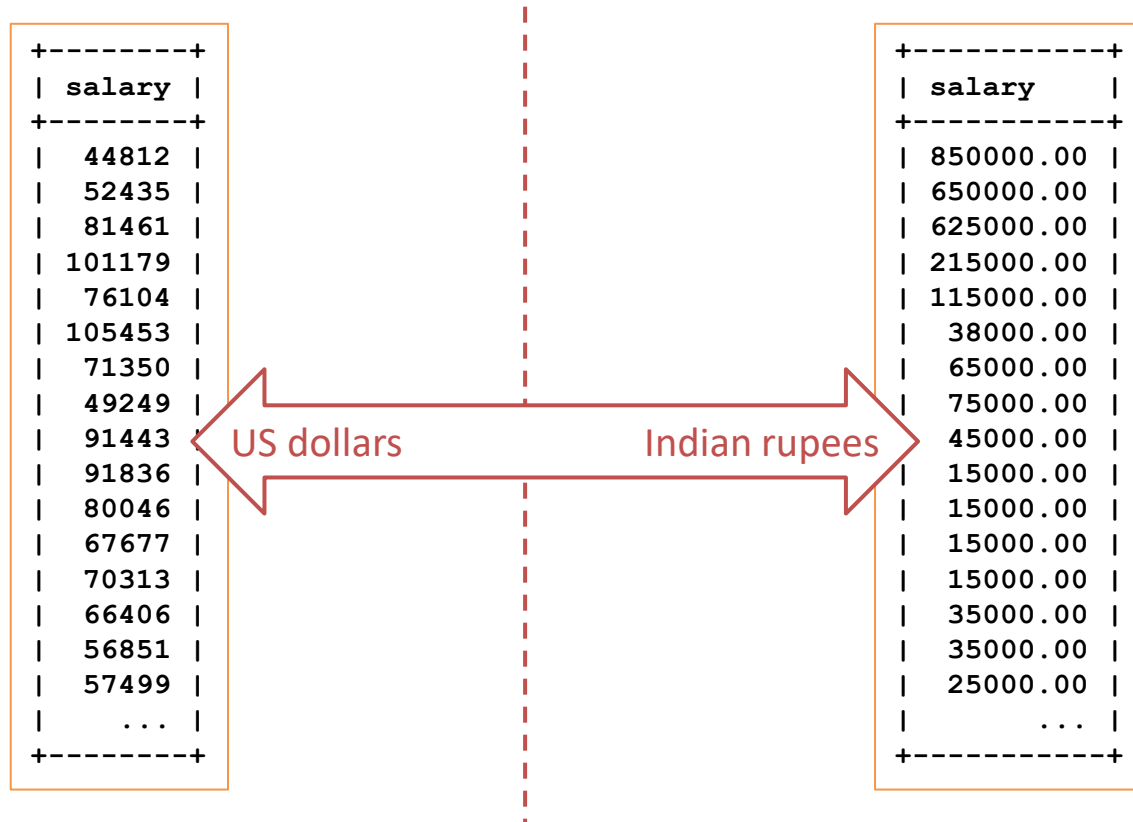
| dept_no | dept_name          |
|---------|--------------------|
| d009    | Customer Service   |
| d005    | Development        |
| d002    | Finance            |
| d003    | Human Resources    |
| d001    | Marketing          |
| d004    | Production         |
| d006    | Quality Management |
| d008    | Research           |
| d007    | Sales              |

| departmentid | departmentname |
|--------------|----------------|
| 101          | IT             |
| 102          | HR             |
| 103          | Finance        |
| 104          | Sales          |
| 105          | marketing      |

– duplicate department names need to be merged

# Data matching – conversion

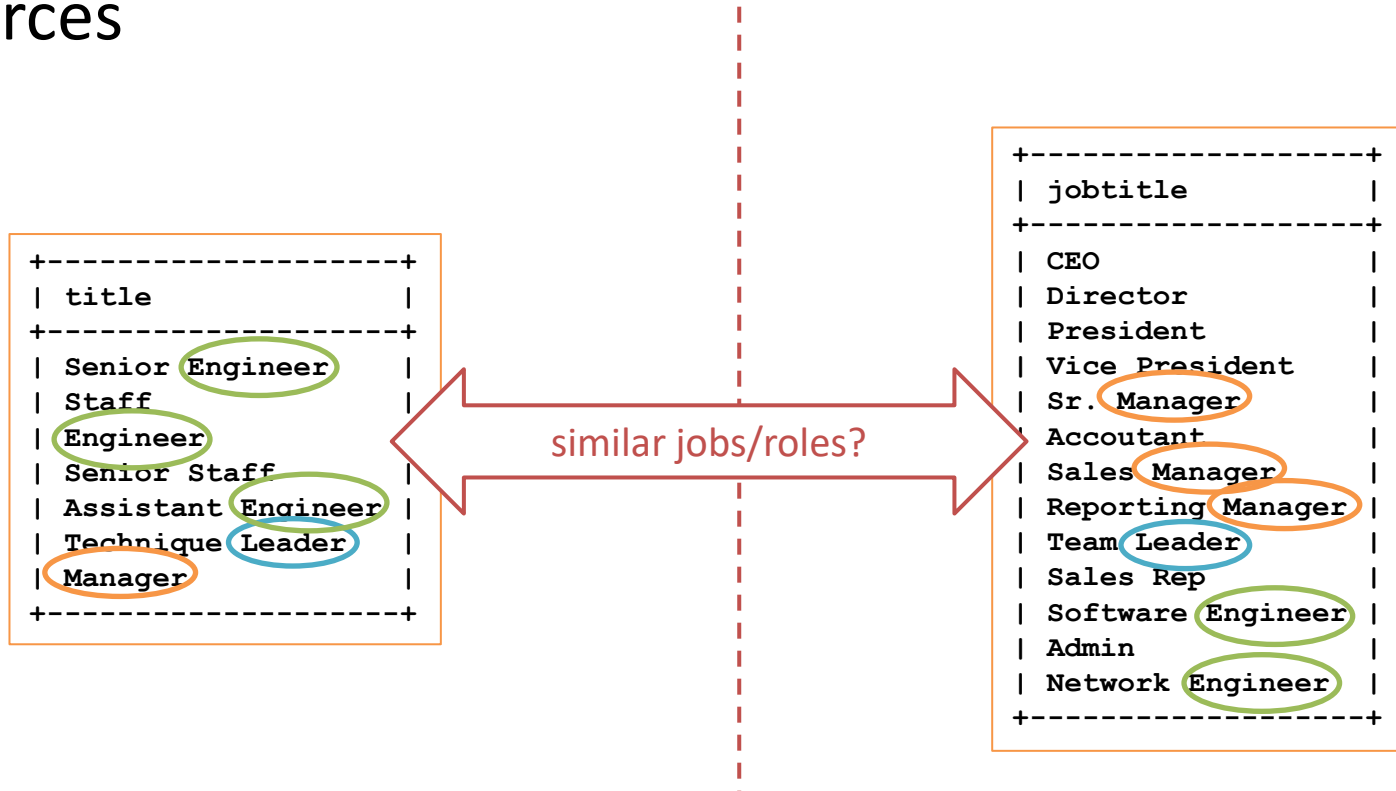
- When comparing data instances from distinct data sources



– salaries need to be converted

# Data matching – approximate duplicates

- When comparing data instances from distinct data sources



– similar job titles need to be found and merged/consolidated



# Summary of concepts

- Multiple **data sources** with different schemas
  - relational databases, but could be other data sources as well
- **Schema matching** between data sources
  - how attributes in one data source correspond to attributes in another data source
- Design of a common **mediated schema**
  - subset of attributes from data source schemas
- **Wrappers** for data sources
  - facilitate and simplify access to data sources
- **Schema mapping** from data sources to mediated schema
  - queries to bring data from local schema to global mediated schema
- **Data matching** between data sources
  - find exact/approximate duplicates from different sources that may need to be merged, converted or consolidated