Imperial College
London

CO130: Databases
2018-02-23

SQL

Tutorial I

Valentin CLÉMENT - vec16@ic.ac.uk

Department of Computing

Accessing the test server

You should all have read access to the following PostgreSQL database:

Host : db.doc.ic.ac.uk

Port : 5432

DB : vec16

Username : *Your college username*

Password : Your ***Postgres*** password, which is ***not*** your college password! You can find it in your emails as **Your new PostgreSQL password and database for [...]**

In this DB, you only have access to selected *schemas*, such as *doc*.

Imperial College London

The DoC teaching database (1/2)

It has been partially replicated into *doc* schema (***FKs*** refer to integer *id*):

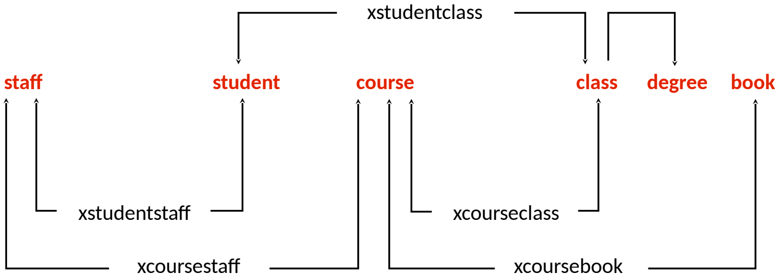
- staff(login, email, lastname, firstname, telephone, room, deptrole, department)
- student(login, email, lastname, status, entryyear, externaldept)
- course(code, title, syllabus, term, classes, popestimate)
- class(***degreeid***, yr, degree, degreeyr, major, majoryr, letter, letteryr)
- degree(title, code, major, grp, letter, years)
- book(code, title, authors, publisher)

Many-to-Many Joining Relations:

- xcourseclass(***courseid***, ***classid***, required, examcode)
- xcoursebook(***courseid***, ***bookid***, rating)
- xcoursestaff(***courseid***, ***staffid***, staffhours, role, term)
- xstudentclass(***studentid***, ***classid***)
- xstudentstaff(***studentid***, ***staffid***, role, grp, projecttitle)

Imperial College London

The DoC teaching database (2/2)



Example 1

- List all staff who do not have a College or Department email address, sort results by last name.
- List all staff with the same last name, show names of staff and their namesake(s).

Relevant tables:

staff (login, email, lastname, firstname, telephone, room, deptrole, department)

Solution 1

- List all staff who do not have a College or Department email address, sort results by last name.

```
select id, lastname, email
from   doc.staff
where  not(email like ’%imperial.ac.uk’
           or email like ’%doc.ic.ac.uk’)
order by lastname
```

Solution 1

- List all staff who do not have a College or Department email address, sort results by last name.

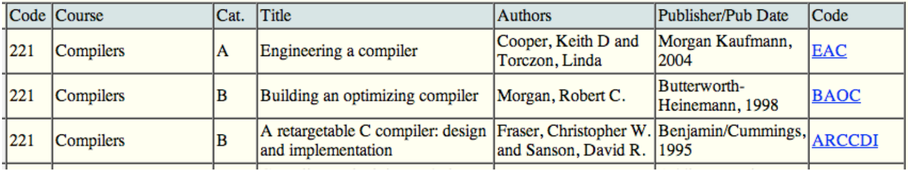
```
select  id, lastname, email
from    doc.staff
where   not(email like ’%imperial.ac.uk’
          or email like ’%doc.ic.ac.uk’)
order by lastname
```

- List all staff with the same last name, show names of staff and their namesake(s).

```
select    s1.id, s1.firstname, s1.lastname,
          s2.id, s2.firstname, s2.lastname
from      doc.staff s1 join doc.staff s2
          on  s1.lastname=s2.lastname
          and s1.id < s2.id
order by s1.lastname
```

Example 2

List all books recommended for courses taught by Prof Kelly, similar to:



Code	Course	Cat.	Title	Authors	Publisher/Pub Date	Code
221	Compilers	A	Engineering a compiler	Cooper, Keith D and Torczon, Linda	Morgan Kaufmann, 2004	EAC
221	Compilers	B	Building an optimizing compiler	Morgan, Robert C.	Butterworth-Heinemann, 1998	BAOC
221	Compilers	B	A retargetable C compiler: design and implementation	Fraser, Christopher W. and Sanson, David R.	Benjamin/Cummings, 1995	ARCCDI

Relevant tables:

staff (login, email, lastname, firstname, telephone, room, deptrole, department)

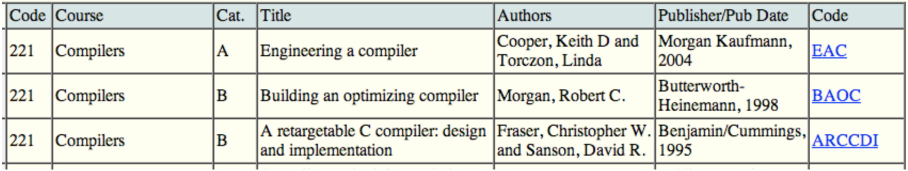
books (code, title, authors, publisher)

course (code, title, syllabus, term, classes, popestimate)

xcoursestaff (***courseid***, ***staffid***, staffhours, role, term)

xcoursebook (***courseid***, ***bookid***, rating)

Solution 2

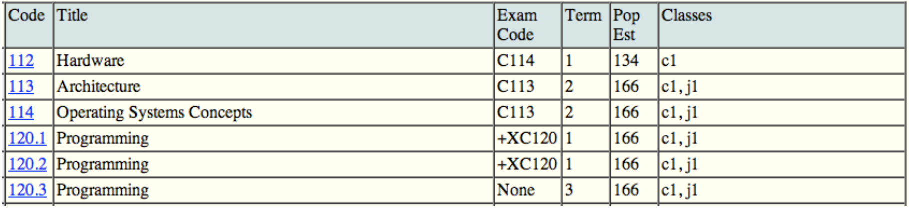


Code	Course	Cat.	Title	Authors	Publisher/Pub Date	Code
221	Compilers	A	Engineering a compiler	Cooper, Keith D and Torczon, Linda	Morgan Kaufmann, 2004	EAC
221	Compilers	B	Building an optimizing compiler	Morgan, Robert C.	Butterworth-Heinemann, 1998	BAOC
221	Compilers	B	A retargetable C compiler: design and implementation	Fraser, Christopher W. and Sanson, David R.	Benjamin/Cummings, 1995	ARCCDI

```
select c.code, c.title , xb.rating , b.title ,
       b.authors , b.publisher , b.code
from   doc.staff s
join   doc.xcoursestaff xc on s.id=xc.staffid
join   doc.course c on xc.courseid=c.id
join   doc.xcoursebook xb on c.id=xb.courseid
join   doc.book b on xb.bookid=b.id
where  s.lastname='Kelly'
order by c.code, xb.rating
```

Example 3

List courses being taken by student with login ***rf611***, similar to:



Code	Title	Exam Code	Term	Pop Est	Classes
112	Hardware	C114	1	134	c1
113	Architecture	C113	2	166	c1, j1
114	Operating Systems Concepts	C113	2	166	c1, j1
120.1	Programming	+XC120	1	166	c1, j1
120.2	Programming	+XC120	1	166	c1, j1
120.3	Programming	None	3	166	c1, j1

Relevant tables:

student (login, email, lastname, status, entryyear, externaldept)

course (code, title, syllabus, term, classes, popestimate)

class (***degreeid***, yr, degree, degreeyr, major, majoryr, letter, letteryr)

xcourseclass (***courseid***, ***classid***, required, examcode)

xstudentclass (***studentid***, ***classid***)

- A direct solution:

```
select c.code, c.title , xc.examcode, c.term,
       c.popestimate, c.classes
from   doc.student s
join   doc.xstudentclass xa on s.id=xa.studentid
join   doc.class ca         on xa.classid=ca.id
join   doc.xcourseclass xc  on ca.id=xc.classid
join   doc.course c         on xc.courseid=c.id
where  s.login='rf611'  order by c.code
```

- A direct solution:

```
select c.code, c.title, xc.examcode, c.term,
       c.popestimate, c.classes
from   doc.student s
join   doc.xstudentclass xa on s.id=xa.studentid
join   doc.class ca         on xa.classid=ca.id
join   doc.xcourseclass xc  on ca.id=xc.classid
join   doc.course c         on xc.courseid=c.id
where  s.login='rf611'  order by c.code
```

- A neater solution:

```
select c.code, c.title, xc.examcode, c.term,
       c.popestimate, c.classes
from   doc.student s
join   doc.xstudentclass xa on s.id=xa.studentid
join   doc.xcourseclass xc  on
           xa.classid=xc.classid
join   doc.course c         on xc.courseid=c.id
where  s.login='rf611'  order by c.code
```

Example 4

- List all PPT tutors and their PPT tutees (*role='PPT'*) as Tutor lastname, Tutor firstname, Tutee lastname, Tutee firstname, sorted
- List all PPT tutors and how many PPT tutees they have, suitably sorted

Relevant tables:

staff (login, email, lastname, firstname, telephone, room, deptrole, department)

student (login, email, lastname, status, entryyear, externaldept)

xstudentstaff (***studentid***, ***staffid***, role, grp, projecttitle)

Solution 4 - i

List all PPT tutor and their tutees:

```
select s.id, s.lastname, s.firstname, t.id,
t.lastname, t.firstname
from   doc.staff s
join   doc.xstudentstaff x on s.id=x.staffid
join   doc.student t on t.id=x.studentid
where  x.role='PPT'
order by s.lastname, s.firstname,
t.lastname, t.firstname
```

Solution 4 - ii

List all PPT tutors and how many PPT tutees they have, suitably sorted:

```
select s.id, s.lastname, s.firstname,
       count(t.id)
from   doc.staff s
join   doc.xstudentstaff x on s.id=x.staffid
join   doc.student t on t.id=x.studentid
where  x.role='PPT'
group by s.id, s.lastname, s.firstname
order by s.lastname, s.firstname
```

Example 5

- List all tutoring roles
- List all tutoring roles and how many tutors there are for each role

Relevant tables:

xstudentstaff (***studentid***, ***staffid***, role, grp, projecttitle)

Solution 5

- List all tutoring roles

```
select role
from   doc.xstudentstaff
group by role
order by role
```

Solution 5

- List all tutoring roles

```
select role
from   doc.xstudentstaff
group by role
order by role
```

- Or alternatively:

```
select distinct role
from   doc.xstudentstaff
order by role
```

Solution 5

- List all tutoring roles

```
select role
from   doc.xstudentstaff
group by role
order by role
```

- Or alternatively:

```
select distinct role
from   doc.xstudentstaff
order by role
```

- List all tutoring roles and how many tutors there are for each role:

```
select role, count(distinct staffid) as tutors
from   doc.xstudentstaff
group by role
order by role
```