Informatics 1 Data & Analysis
Tutorial 3

Week 5, Semester 2, 2012–2013

- You must prepare for the tutorial by attempting the questions on this worksheet in advance. Bring with you a copy of your work, including printouts of code and other results.

  If you cannot do some questions, write down what it is that you find challenging and use this to ask your tutor in the meeting.

  It's important both for your learning and other students in the group that you come to tutorials properly prepared. If you have not attempted the exercise sheet, then you may be sent away from the tutorial to do it elsewhere.
- Some exercise sheets contain material marked with a star ⋆. These are optional extensions.
- Data & Analysis tutorial exercises are not assessed, but they are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.
- Attendance at tutorials is obligatory: if you are ill or otherwise unable to attend one week then email your course tutor, and if possible attend another tutorial group in the same week.
- *Related Reading:* Chapter 4 (Relational Algebra and Calculus) of *Database Management Systems*, Raghu Ramakrishnan and Johannes Gehrke, 2003.

Introduction

In this tutorial, you will construct queries in *tuple-relational calculus* and describe operations to compute their results using *relational algebra*. These systems were introduced, with examples, in the lectures. All questions are based on a set of relational tables dealing with air travel: airports, flights, bookings and seats. This tutorial is a bit more difficult than the previous ones. If you need any help, please ask the drop-in lab demonstrators. If you are stuck with any question, write down what you are finding difficult and then move on to try the next one.

A Relational Model for Flight Bookings

The following DDL declarations and table data describe a relational model for air travel bookings. This is a very simplified model — in particular, we don't deal with times or dates of individual flights. Notice that we are taking advantage of SQL's case insensitivity for keywords, with **create table** and **primary key** instead of **CREATE TABLE** and **PRIMARY KEY**.

```
create table Airport (
    airportId      varchar(3),
    name           varchar(50),
    city           varchar(30),
    primary key (airportId)
)
```

```
create table Flight (
    flightNo      varchar(6),
    flightCompany varchar(20),
    depAirport    varchar(3),
    arrAirport    varchar(3),
    primary key (flightNo),
    foreign key (depAirport) references Airport(airportId),
    foreign key (arrAirport)  references Airport(airportId)
)

create table Booking (
    ticketNo      varchar(9),
    name          varchar(20),
    nationality   varchar(20),
    flightNo      varchar(6),
    seatNo        varchar(3),
    primary key (ticketNo),
    foreign key (flightNo)  references Flight,
    foreign key (seatNo)    references Seat
)

create table Seat (
    seatNo        varchar(3),
    flightNo      varchar(6),
    class         varchar(10),
    primary key (seatNo, flightNo),
    foreign key (flightNo) references Flight
)
```

Airport

airportId	name	city
LHR	Heathrow	London
LGW	Gatwick	London
CDG	Charles de Gaulle	Paris
ORY	Orly	Paris

Flight

flightNo	flightCompany	depAirport	arrAirport
AF1231	Air France	LHR	CDG
AF1232	Air France	CDG	LHR
AF1234	Air France	LGW	CDG
AF1235	Air France	CDG	LGW
BA2943	British Airways	LGW	ORY
BA2944	British Airways	ORY	LGW
BA4059	British Airways	LHR	CDG
BA4060	British Airways	CDG	LHR

Booking

ticketNo	name	nationality	flightNo	seatNo
EAG129489	John Jones	British	AF1232	12D
EAF123456	Fraser McEwan	British	AF1232	30E
ABS958332	Jane Smith	French	BA2944	10A
ORE394895	Fiona Stewart	British	BA4060	5D
EYR149583	Tom Woods	British	BA4059	14B
EAG348595	John Smith	French	BA2944	30D

Seat

seatNo	flightNo	class
12D	AF1232	Business
30E	AF1232	Economy
10A	BA2944	Business
5D	BA4060	Business
14B	BA4059	Economy
30D	BA2944	Economy

Part 1: Operations in Relational Algebra

For each of the following queries in relational algebra, calculate the output table and give a brief statement of what query it answers.

(a) $\sigma_{\mathsf{class='Business'}}(\mathsf{Seat})$

(b) $\pi_{\mathsf{nationality}}(\mathsf{Booking})$

(c) $\sigma_{\mathsf{nationality='French'}}(\mathsf{Booking}) \times \sigma_{\mathsf{class='Business'}}(\mathsf{Seat})$

(d) $\mathsf{Booking} \bowtie \mathsf{Seat}$

(e) $\pi_{\mathsf{name}}(\sigma_{\mathsf{class='Business'}}(\mathsf{Booking} \bowtie \mathsf{Seat}))$

(f) $\mathsf{Airport} \cup \mathsf{Seat}$

Part 2: Constructing Queries

For each of the following questions, formulate the specified queries in tuple-relational calculus and as a computation in relational algebra.

(a) Retrieve all information about airports in London. The schema of the output table should be same as that of the Airport table.

(b) Retrieve details of all bookings by British and French passengers. The schema of the output table should be same as that of the Booking table.

(c) Retrieve the names of all passengers.

(d) Retrieve the flight number, Departure and Arrival airports of all British Airways flights.

(e) Retrieve the name of every passenger together with their flight number and the associated flight company.

⋆ **(f)** Retrieve details of all flights from all airports in London. The output schema should be same as that of Flight table.

⋆ **(g)** Find out the ticket numbers and names of all passengers departing from London.

⋆ **(h)** Retrieve the flight number and company of all flights from London to Paris.