Informatics 1: Data & Analysis

Lecture 21: Exam Preparation

Ian Stark

School of Informatics
The University of Edinburgh

Tuesday 29 March 2016
Semester 2 Week 11

Plan !

In this lecture I shall work through solutions to two past exam questions.

August 2014	Question 1
May 2015	Question 2

As well as the usual slide handout, I've included extracts from the feedback reports written after each exam. These contain further notes on solutions and the different answers given by students in the exam itself.

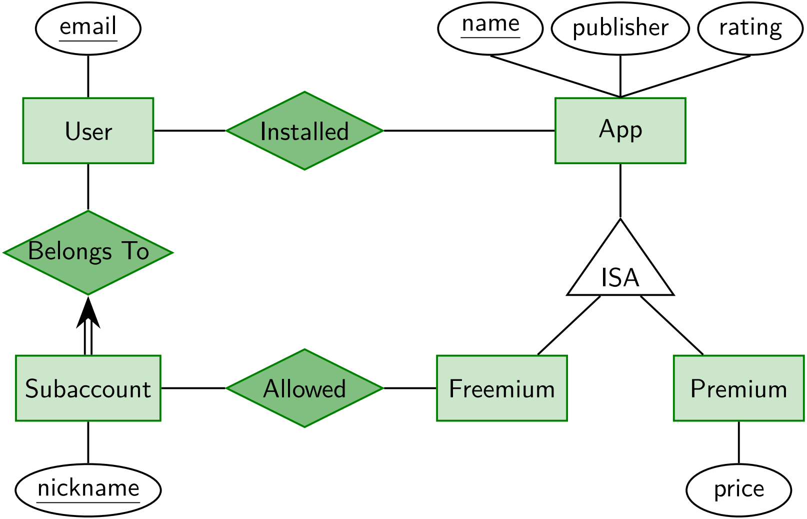
August 2014 Question 1 Section (a)

A phone company wants to set up their own App Store for mobile devices. Requirements analysis for the controlling database highlights the following information about what must be recorded.

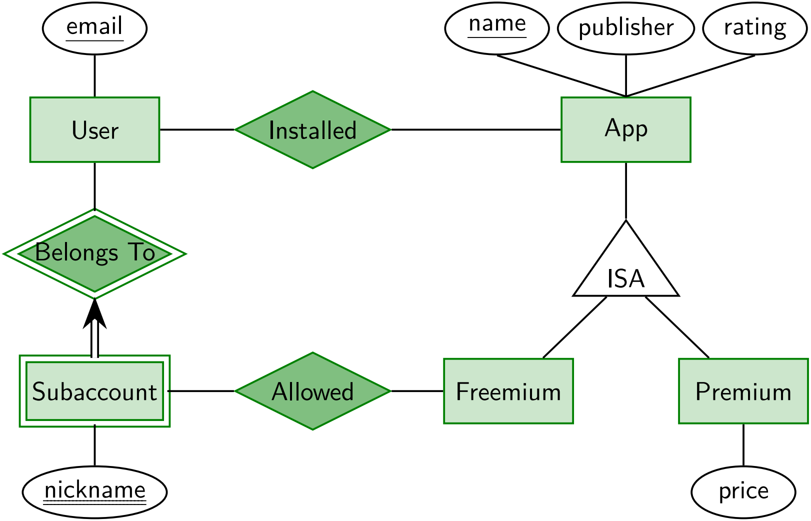
- Every app in the store needs a unique name, a publisher, and a rating.
- There are two subclasses of app: a *premium* app has to be paid for before installation; a *freemium* app is free to download, but has in-app purchases which cost money.
- The database should record the price of each premium app.
- Each user of the store is identified by their email address.
- A user may have several subaccounts, each identified by a nickname.
- The database needs to record which users have installed which apps.
- Users can use subaccounts to restrict access to freemium apps: the database needs to record which nicknames are allowed to run which ones.

Draw an entity-relationship diagram to represent this information.

August 2014 Question 1 Section (a)



August 2014 Question 1 Section (a)



August 2014 Question 1 Section (a)

Using a weak entity is reasonable but not essential. Some other features:

- The name of an app is unique, so there is no need to include publisher in the key.
- Total participation (double line) of SubAccount in BelongsTo: every subaccount must belong to some user.
- Key constraint (arrowhead) between SubAccount and BelongsTo: every subaccount can belong to at most one user.
- No arrowheads around the Installed relationship: each user may install many apps, and each app may be installed by many users.
- No arrowheads around the Allowed relationship: each subaccount may be authorised for many freemium apps, and each freemium app may be available to multiple subaccounts.

August 2014 Question 1 Section (b)

Q. What do the terms “arity” and “cardinality” mean when describing database tables?

A. The *arity* of a database table is the number of columns (fields, attributes) it has. The *cardinality* of a database table is the number of rows (tuples, records) it contains.

August 2014 Question 1 Sections (c)–(e)

The app store groups apps into *themes* such as “Games”, “News + Magazines”, or “Health + Fitness”. An app can be in multiple themes, and each theme can have a current “Top App”. This is captured by the following SQL data declarations.

```
create table App (
  name       varchar(30),
  publisher  varchar(25),
  rating     integer,
  primary key (name)
)
```

```
create table Theme (
   title     varchar(20),
  topApp     varchar(30),
  primary key  ( title ),
  foreign key   (topApp) references App(name)
)

create table InTheme (
  name  varchar(30),
   title  varchar(20),
  primary key  (name,title),
  foreign key   (name)  references App,
  foreign key   ( title )   references Theme
)
```

August 2014 Question 1 Sections (c)–(e)

(c) Write relational algebra expressions to compute the following.

  (i) The name of the top app in the "Games" theme.

  (ii) For every app in the "Games" theme, its name and rating.

(d) Write expressions in the tuple-relational calculus that express the following queries.

  (i) The names of all apps in the "Office" theme.

  (ii) The publishers of all top apps.

(e) Write SQL queries to answer the following questions.

  (i) How many apps are there in the database?

  (ii) What is the highest and lowest rating given to apps in the "Utilities" theme?

August 2014 Question 1 Sections (c)–(e)

(c) (i) $\pi_{\text{topapp}}(\sigma_{\text{title}='\text{Games}'}(\text{Theme}))$

(ii) Either $\pi_{\text{name,rating}}(\sigma_{\text{title}='\text{Games}'}(\text{InTheme}) \bowtie \text{App})$

or $\pi_{\text{name,rating}}(\sigma_{\text{title}='\text{Games}'}(\text{InTheme} \bowtie \text{App}))$

(d) (i) $\{ R \mid \exists X \in \text{InTheme} \,.\, X.\text{title} = '\text{Office}' \wedge X.\text{name} = R.\text{name} \}$

(ii) $\{ R \mid \exists T \in \text{Theme}, A \in \text{App} \,.\, T.\text{topApp} = A.\text{name}$
$\wedge\ A.\text{publisher} = R.\text{publisher} \}$

(e) (i) **select count**$(*)$ **from** App

select count(name) **from** App

(ii) **select min**(rating), **max**(rating)
from App, InTheme
where App.name = InTheme.name **and** InTheme.title = 'Utilities'

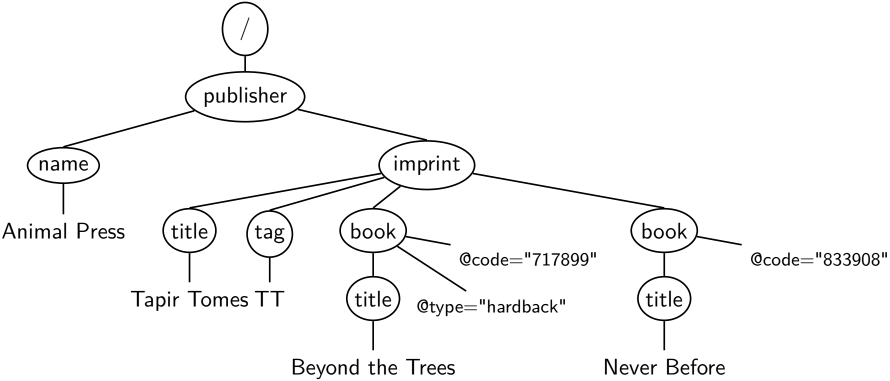
May 2015 Question 2 Section (a)

This XML document captures some information about a book publisher's catalogue: in this case, the non-existent publisher *Animal Press*.

Draw the tree of the *XPath data model* for this XML document.

```
<?xml version="1.0"?>
<!DOCTYPE publisher SYSTEM "publisher.dtd">
<publisher>
    <name>Animal Press</name>
    <imprint>
        <title>Tapir Tomes</title>
        <tag>TT</tag>
        <book code="717899" type="hardback">
            <title>
                Beyond the Trees
            </title>
        </book>
        <book code="833908">
            <title>
                Never Before
            </title>
        </book>
    </imprint>
</publisher>
```

May 2015 Question 2 Section (a)



May 2015 Question 2 Section (b)

A full catalogue would list many different *imprints* of the publisher — different brand names they use to sell books — with for each imprint a short unique *tag* and a list of many books. Every book has a unique code, and is either *hardback* or *paperback*, with the default being paperback if not specified.

Write out a DTD which describes this document and any other similar publisher's catalogue, suitable for the "publisher.dtd" file referenced.

May 2015 Question 2 Section (b)

```
<!ELEMENT publisher (name,imprint+) >
<!ELEMENT imprint (title,tag,book+) >
<!ELEMENT book (title) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT title (#PCDATA) >
<!ELEMENT tag (#PCDATA) >
<!ATTLIST book code CDATA #REQUIRED>
<!ATTLIST book type (hardback|paperback) "paperback" >
```

- The order of lines does not matter.
- Only one declaration of title element, although used in two different ways (title of a book, title of an imprint).

May 2015 Question 2 Section (c)

Write XPath expressions to obtain the following information from such a document.

(i) A list of all the imprint tags.

(ii) The title of the book with code 823095.

(iii) The title of every imprint that includes at least one hardback book.

May 2015 Question 2 Section (c)

(i) //tag/text()

/publisher/imprint/tag/text()

//imprint/tag/text()

(ii) //book[@code="823095"]/title/text()

//title [../ @code="823095"]/text()

(iii) //imprint[book/@type="hardback"]/title/text()

//imprint[.//@type="hardback"]/title/text()

//book[@type="hardback"]/../title/text()

May 2015 Question 2 Sections (d) and (e)

(d) The *Animal Press* themselves keep this information in a relational database with two linked tables: Imprint and Book. Write suitable schemas for these tables in the SQL Data Declaration Language.

(e) Based on your schemas, write SQL queries to find out the information required for each item in part (d) above.

May 2015 Question 2 Section (d)

```
create table Imprint (
    title  varchar(60) not null,
    tag    varchar(6),
    primary key (tag)
)

create table Book (
    title      varchar(120) not null,
    code       varchar(6),
    type       varchar(10),
    imprint    varchar(6) not null,
    primary key (code),
    foreign key (imprint) references Imprint(tag)
)
```

May 2015 Question 2 Section (e)

(i)

```
select tag from Imprint
```

(ii)

```
select title from Book where code = '823095'
```

(iii)

```
select distinct Imprint.title
from Imprint, Book
where Book.imprint=Imprint.tag and Book.type='hardback'
```

- Use **distinct** as imprint may include multiple hardback books.
- Fully-qualified Imprint.title as Book also has title field.

Rocket Science

Course Survey !

Please complete the online survey for this course. This is anonymous, and I read every submission.

The survey should go live this afternoon: do it then.

MyEd ⟶ Course Surveys

Experimental direct URL: **http://is.gd/infsurvey**

If that URL works for you, send me
a screenshot of where it takes you

Finis !

You can do this

The Inf1-DA syllabus and exam questions are written to be achievable. Every year large numbers of students pass the exam writing straightforward correct answers about things they understand. You can do this too.

Anything Else?

If you have further questions about the course content, tutorial exercises, the exam, where to buy a disco calculator, or anything else, please:

- Post a question on *Piazza*; *or*
- Ask your course tutor, in person or by email; *or*
- Ask me, in person or by email.

Thank you for your attention

We're done here