COS 326
Database Systems

Lecture 1
Admin matters
Introduction: Course overview
21 July 2015
Admin matters

- All details are available on COS326 website
- Lecturers and teaching assistants:

<table>
<thead>
<tr>
<th>Name</th>
<th>Capacity</th>
<th>Office</th>
<th>Contact details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Patricia E.N. Lutu</td>
<td>Lecturer &amp; course coordinator</td>
<td>IT 5-41</td>
<td><a href="mailto:plutu@cs.up.ac.za">plutu@cs.up.ac.za</a></td>
</tr>
<tr>
<td>still to be finalised</td>
<td>Teaching Assistant</td>
<td></td>
<td>xt 4116</td>
</tr>
<tr>
<td>still to be finalised</td>
<td>Teaching Assistant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>still to be finalised</td>
<td>Teaching Assistant</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Module Time table:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Day</th>
<th>Time</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture 1</td>
<td>Tuesday</td>
<td>12:30 – 13:20</td>
<td>IT 4-2</td>
</tr>
<tr>
<td>Lecture 2</td>
<td>Wednesday</td>
<td>13:30 – 14:20</td>
<td>IT 2-24</td>
</tr>
<tr>
<td>Practical session</td>
<td>Friday</td>
<td>12:30 – 15:20</td>
<td>Red lab, Informatorium</td>
</tr>
</tbody>
</table>
Study material: Theory

Textbook:
Thomas Connolly and Carolyn Begg
*Database Systems - A Practical Approach to Design, Implementation, and Management*
Addison Wesley, 5th edition 2010,
ISBN-10 0-321-52306-7

(**price of textbook**)
Study material: for practicals

**Software** will be made available on:
1. the ftp site of the department at [ftp.cs.up.ac.za](ftp.cs.up.ac.za)
2. and/or through links on the CS web site at [http://www.cs.up.ac.za/courses/COS326](http://www.cs.up.ac.za/courses/COS326)

<table>
<thead>
<tr>
<th>Software</th>
<th>Available from</th>
</tr>
</thead>
<tbody>
<tr>
<td>db4objects</td>
<td></td>
</tr>
<tr>
<td>BaseX</td>
<td><a href="http://basex.org">http://basex.org</a></td>
</tr>
<tr>
<td>MongoDB</td>
<td><a href="http://www.mongodb.org/downloads">http://www.mongodb.org/downloads</a></td>
</tr>
<tr>
<td>Neo4j</td>
<td><a href="http://www.neo4j.org/download">http://www.neo4j.org/download</a></td>
</tr>
</tbody>
</table>
## Course work and assessment

### Project days???

<table>
<thead>
<tr>
<th>Activity</th>
<th>Number of assessments</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Practical work</td>
<td>8</td>
<td>individual</td>
</tr>
<tr>
<td>(all in 1&lt;sup&gt;st&lt;/sup&gt; half of semester)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Class tests</td>
<td>3</td>
<td>individual</td>
</tr>
<tr>
<td>3. Theory assignments &amp; class presentations</td>
<td>1 per group</td>
<td>group</td>
</tr>
<tr>
<td>4. Assessment of other groups</td>
<td>14 per group</td>
<td>group</td>
</tr>
<tr>
<td>5. Semester test</td>
<td>1</td>
<td>individual</td>
</tr>
<tr>
<td>6. Exam</td>
<td>1</td>
<td>individual</td>
</tr>
</tbody>
</table>
(Group) theory assignments

- Work in **groups of four or five (max 15 groups)**
- Select one of the assignment topics by **Friday 7 August**.
- Dates for the presentations are indicated in study guide.
- Hand in the assignment on the Friday before the presentation date.
- During the second part of the lecture that you selected, you will give a **10 minutes presentation** of your assignment.

- **Refer to the following documents for details:**
  1. COS326-Assignments-2015.pdf
  3. Study guide section 8.2 for presentation schedule
COS326 website & Study guide

• Refer to study guide (on website) for:
  (1) all details on lecturers, textbook, study materials
  (2) assessment & mark allocation
  (3) theory assignments
  (4) course timetable

• Refer to website for:
  (1) practical exercises
  (2) Bookings for marking of pracs
  (3) Bookings for assignment topics & dates
  (4) Marksheets
  (5) Date for semester test
# Course overview (1)

This module builds on prior introductory modules on database systems and provides more advanced theoretical and practical study material.

- **Prerequisites**
  - INF 214, *Databases and database design* or permission from the Head of Department

- **Related modules**
  - INF 214: *Databases and database design*
  - INF 261: *Database management* Software Development elective group in Bsc (IT) are required to take this module in the second year.
  - **COS787: Spatial Databases (Honours)**
    - (COS326 is a pre-requisite)
Course overview (2)

- **Database models**
  - Review of relational databases (revision)
  - Object-oriented databases
  - Object-relational databases
  - Spatial databases
  - Semi-structured databases (XML)
  - NoSQL databases

- **Advanced topics**
  - Data analytics (Data warehousing, OLAP, data mining)
  - Analytics for Big Data
Relational databases

• **Relation**
  - a table with columns and rows
    • applies to logical structure of the database
    • NOT the physical structure

• **Relational database**
  - a collection of *normalized relations* with distinct relation names
  - is based on *a schema*
Instances of branch and staff relations

<table>
<thead>
<tr>
<th>branchNo</th>
<th>street</th>
<th>city</th>
<th>postcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>B005</td>
<td>22 Deer Rd</td>
<td>London</td>
<td>SW1 4EH</td>
</tr>
<tr>
<td>B007</td>
<td>16 Argyll St</td>
<td>Aberdeen</td>
<td>AB2 3SU</td>
</tr>
<tr>
<td>B003</td>
<td>163 Main St</td>
<td>Glasgow</td>
<td>G11 9QX</td>
</tr>
<tr>
<td>B004</td>
<td>32 Manse Rd</td>
<td>Bristol</td>
<td>BS99 1NZ</td>
</tr>
<tr>
<td>B002</td>
<td>56 Clover Dr</td>
<td>London</td>
<td>NW10 6EU</td>
</tr>
</tbody>
</table>

**Domain:** set of allowable values for one or more attributes

**Entity integrity,** **Referential integrity**

<table>
<thead>
<tr>
<th>staffNo</th>
<th>fName</th>
<th>IName</th>
<th>position</th>
<th>sex</th>
<th>DOB</th>
<th>salary</th>
<th>branchNo</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL21</td>
<td>John</td>
<td>White</td>
<td>Manager</td>
<td>M</td>
<td>1-Oct-45</td>
<td>30000</td>
<td>B005</td>
</tr>
<tr>
<td>SG37</td>
<td>Ann</td>
<td>Beech</td>
<td>Assistant</td>
<td>F</td>
<td>10-Nov-60</td>
<td>12000</td>
<td>B003</td>
</tr>
<tr>
<td>SG14</td>
<td>David</td>
<td>Ford</td>
<td>Supervisor</td>
<td>M</td>
<td>24-Mar-58</td>
<td>18000</td>
<td>B003</td>
</tr>
<tr>
<td>SA9</td>
<td>Mary</td>
<td>Howe</td>
<td>Assistant</td>
<td>F</td>
<td>19-Feb-70</td>
<td>9000</td>
<td>B007</td>
</tr>
<tr>
<td>SG5</td>
<td>Susan</td>
<td>Brand</td>
<td>Manager</td>
<td>F</td>
<td>3-Jun-40</td>
<td>24000</td>
<td>B003</td>
</tr>
<tr>
<td>SL41</td>
<td>Julie</td>
<td>Lee</td>
<td>Assistant</td>
<td>F</td>
<td>13-Jun-65</td>
<td>9000</td>
<td>B005</td>
</tr>
</tbody>
</table>

© Pearson Education Limited 1995, 2005
Object-Oriented & Object-Relational databases

- **Object-oriented programming**
  - allows software structure to be based on *real-world structures*, and gives programmers a powerful way to *simplify the design and construction* of complex programs.

- **Object-oriented DB**
  - stores objects
  - does not use tables (relations)
Weaknesses of RDBMSs (Motivation for OODB & ORDB)

- Poor representation of “real world” entities
  - Normalization leads to relations that do not correspond to entities in the “real world”
  - Fragmentation of the “real world”

<table>
<thead>
<tr>
<th>Staff</th>
<th>Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>staffNo</td>
<td>sName</td>
</tr>
<tr>
<td>SL21</td>
<td>John White</td>
</tr>
<tr>
<td>SG37</td>
<td>Ann Beech</td>
</tr>
<tr>
<td>SG14</td>
<td>David Ford</td>
</tr>
<tr>
<td>SA9</td>
<td>Mary Howe</td>
</tr>
<tr>
<td>SG5</td>
<td>Susan Brand</td>
</tr>
<tr>
<td>SL41</td>
<td>Julie Lee</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>branchNo</th>
<th>bAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>B005</td>
<td>22 Deer Rd, London</td>
</tr>
<tr>
<td>B007</td>
<td>16 Argyll St, Aberdeen</td>
</tr>
<tr>
<td>B003</td>
<td>163 Main St, Glasgow</td>
</tr>
</tbody>
</table>

(expensive) JOIN required to see branch address together with staff member’s name
Object Relational Database

- extend relational model with OO features
  - objects, classes, encapsulation
  - inheritance, polymorphism, etc…
  - => reduces the need for JOINS

- Bridges the gap between
  - Programming language and RDBMS data
    - can write code to access a DB table of objects
      (without having to join tables)

- Object Oriented data modeling and RDBMS
  - direct mapping between design and DB implementation
## XML:

### Semi-structured databases

#### Semi-structured data

- irregular or incomplete
- may have a structure, but may change

**schema-less or self-describing**

- no separate schema
- schema is part of the data

### Relational, object-oriented & object-relational DBMSs:

- Require pre-defined schema

---

**e.g. music_cd_catalog.xml**

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<CATALOG>
  <CD>
    <TITLE>Joyous 12</TITLE>
    <ARTIST>Joyous</ARTIST>
    <COUNTRY>RSA</COUNTRY>
    <COMPANY>Xyz</COMPANY>
    <PRICE>150.00</PRICE>
    <YEAR>2012</YEAR>
  </CD>
  <CD>
    <TITLE>In His Presence</TITLE>
    <ARTIST>Benjamin Dube</ARTIST>
    <COUNTRY>RSA</COUNTRY>
    <COMPANY>Xyz</COMPANY>
    <PRICE>120.00</PRICE>
    <YEAR>2010</YEAR>
  </CD>
</CATALOG>
```
NoSQL database systems:

- are distributed non-relational DBs
- designed for large-scale data storage and processing

Who uses NoSQL database systems?

- Arose alongside major Internet businesses which had challenges in storing & processing huge quantities of data: Google, Amazon, Facebook, Twitter, Yahoo!, etc.

- Today they are used by organisations that collect large amounts of unstructured data (for analysis)
NoSQL database systems (5-2)

- **Querying the database**
  - no standard query language
  - DBMS provides:
    - native API with simple data access commands e.g. put, read, delete OR
    - a query language specific to the DBMS

- **Classification of NoSQL databases**
  1. Key-value stores
  2. document databases
  3. wide-column stores
  4. Graph databases
Document databases

- designed to store documents
- documents are encoded in a standard data exchange format
  - e.g. XML, JSON, BSON (Binary JSON)
- data stored in key-value pair style but value column is un-structured data

**Primary use:** storing text documents, e-mail messages, XML documents

**Example DBs:** MongoDB, Apache’s CouchDB (FLOSS)
NoSQL database systems (5-4)

Graph databases

- use structured graphs of interconnected key-value pairings

- similar to OODBs. Graph represented as an OO network of:
  - **nodes** (objects),
  - **edges** (node relationships),
  - **properties** (object attributes expressed as key-value pairings)

**Primary uses:** representing social networks, generating recommendations, conducting forensic investigations

**Example DBs:** Neo4j, InfoGrid, AllegroGraph
Reading for next lecture

• Tomorrow’s lecture and next week Tuesday:
  • Object-Oriented DBMSs
    • Chapter 27
    • Chapter 28
  • Practical exercise 1
    • available on COS326 website on Friday
    • due date Friday 31 July