Intro to Relational Databases

Problem statement

- A database system should provide a general-purpose way for users to structure their data
- Requirements:
  - the data model should be simple and easy to understand
  - it should be easy to see how data for an application should be represented
  - it should allow for easy and efficient data processing

Relational Model

- In 1970s, Codd purposed a DBMS that present the user with view of data organized as tables called relations
  - provide abstraction (low level details)
  - underlying data structures to provide rapid response to variety of queries
  - queries are declarative in nature

A relational DB is a group of tables

Patient No.	Last name	First name	Sex	Date of birth	Ward No.
454	Smith	John	M	14.08.58	6
223	Jones	Peter	M	07.12.65	8
597	Brown	Brenda	F	17.06.61	3
234	Jenkins	Alan	M	29.01.67	7
244	Wells	Christopher	M	25.02.55	6

row

column

Ward No.	Ward name	Type	No. of Beds
3	Carey	Medical	8
6	Bracken	Medical	16
7	Brent	Surgical	12
8	Meavy	Surgical	10

The values in the table are all **simple**: integers, strings, etc.

A table from the 'Courses' DB

A column is also called an "attribute"

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

A row is also called a "tuple".

Another table from the 'Courses' DB

course

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principals	Physics	4

A tuple represents a relationship between the values of the tuple.

A table represents a mathematical relation.

So a table is also called a 'relation'.

This is why we say "relational database".

Another 'Courses' table

Each attribute has a 'domain', which is a set of values that are allowed in the column.

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46

What is another word we could use for 'domain'?

Null values

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	null
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	null	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Sometimes we don't know a value, or the value doesn't exist.

In relational DBs we deal with this by using special 'null' values.

Question: could it be that we didn't know an instructor's salary?

Relation schema and instances

relation schema

gives the names and domains of the attributes (domains not shown here.)

```
student(ID, name, dept_name, tot_cred)
```

relation instance

an instance of a relation schema

also called a 'table'

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46

Database schema and instances

database schema

the relation schemas for all the relations in a database

```
student(ID, name, dept_name, tot_cred)
department(dept_name, building, budget)
…
```

database instance

all the tables of the database

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46

department

Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Engr.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

...

All the names

In databases there seem to be two or three ways to say everything.

schema = relation schema

table = relation = relation instance = instance

tuple = row

attribute = column

XML – data model

- EXtensible Markup Language
  - Exchanging information on the Internet
- HTML vs XML
  - XML tags describe the content of the data rather than format

```
<bookstore>
    <book isbn="isbn-0-1" price=20>
        <title>booktitle</tile>
        <author>foo<author>
    <book>
</bookstore>
```

Relational vs XML

	Relational	XML
Structure	Set of Tables	Hierarchical Tree (nested elements)
Schema	Fixed (defined in advance)	Flexible
Queries	SQL	XPATH, XQUERY
Ordering	None	Implied

JSON – data model

- JavaScript Object Notation
  - semi-structured data for data exchange
  - Independent of Javascript
  - Recursive constructs – objects {}, arrays[]

```
{“books”:
    [  { “isbn”:”isbn-0123”,
         “title”:”booktitle”,
         “author”: [ {“name”:”foo”},
                     {“name”:”bar”}]
```

Relational vs JSON

	Relational	JSON
Structure	Set of Tables	Sets of label pairs and arrays
Schema	Fixed (defined in advance)	Self-describing data
Queries	SQL	JAQL, JSON path/query
Ordering	None	Implied – arrays are ordered

Lab

1. In databases we have data models, schemas, and tables. What are the analogous things in the world of programming languages?
2. Give another name for an attribute of a database table.
3. What's the difference between a table and a relation instance?
4. Would you ever want a table in which two rows were the same?
5. Does the order of rows within a table matter?
6. Does the order of attributes within a table matter?

Lab - Solutions

1. OO programming / classes / objects
2. Column
3. Nothing, they are synonyms
4. No
5. No, If you care about the order when displaying rows in a query, you say that in the query.
6. No, not technically, although it is common to put the primary key attributes as the first attributes in a relation schema.