Advanced Databases

1. Module Introduction

0. Module Introduction

0.1 Timetable

0.2 Assessment Structure

0.3 Reading List

0.4 Q&A

0.1 Timetable

- Lectures
  - □ Monday 3:00pm – 5:00pm SCR1
- Labs
  - □ Thursday 10:00am – 11:00am SCR2

0.2 Assessment Structure

Allocation of marks	
Continuous Assessment	50%
Final Examination	50%
Total	100%

CA Structure & Dates	
CA 1 – In Class Test – Week 6	20%
CA 2 – Research Paper – Week 11	20%
CA 3 – Data- Warehouse Design – Week 12	100%

0.3 Reading List

Required Reading:	Connolly T., Begg C., Strachan A, Database Systems - A Practical Approach to Design, Implementation and Management, Fourth Edition, Addison-Wesley 2005
Supplementary Reading:	Elmasri B., Navathe S., Fundamentals of Database Systems, Fourth Edition, Prentice Hall 2003 Gillenson, M., L., Fundamentals of Database Management Systems, 2005, Wiley Kroenke, D.M., Database Processing, Eight Edition, Prentice Hall 2002 Manning C., Raghaven, P., & Schutze, H., (2008), Introduction to Information Retrieval, Cambridge University Press Rolland F.D., The Essence of Databases, Prentice-Hall 1998 Riccardi G., Principles of Database Systems with Internet and Java Applications, Prentice Hall 2003

0.4 Q&A

Why Data Bases?

- A huge amount of information being stored.
- The College, Medical records, Employers, Companies, Government Agencies etc.
- Managing that data is a mammoth task
- Data Base Management Systems (DBMS)
- Storing is easy, managing is the issue
- A number of models available

Data Models

- Hierarchical
- Network
- Relational
- Object-Oriented
- Distributed Databases

Benefits of Database Approach

- Data can be shared
- Redundancy can be reduced
- Inconsistency can be avoided
- Transaction support can be provided
- Integrity can be maintained
- Security can be enforced
- Conflicting requirements can be balanced
- Standards can be enforced

Disadvantages of Database

- Shared data can be abused
- Controls needed to ensure data quality is maintained
- Data integrity during multi user access must be maintained
- Enterprise vulnerability
- Cost

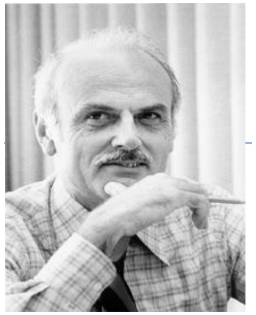
Models

- First Generation
  - File Based DB
  - Hierarchical DB
  - Network DB
- Second Generation
  - Relational DB
- Third Generation
  - Object-Oriented DB
  - Deductive DB
  - Distribution

The Relational Model

- Formulated by Codd in 1970
- Commercial RDBMS in 80s
- 12 Rules specified by Codd
- Most widely used Model at present
  - Access, Oracle, MySQL, SQL Server, Teradata etc

E. F. Codd 1970



- Edgar (Ted) Codd born in Britain, worked for IBM, although Oracle were first to implement his ideas
- Paper : A Relational Model of Data for Large Shared Data Banks
- Data independence was his starting point
- Relation was used in its mathematical sense, i.e. relations between sets of data (domains)
- Introduced normalisation
- Issues of redundancy and consistency

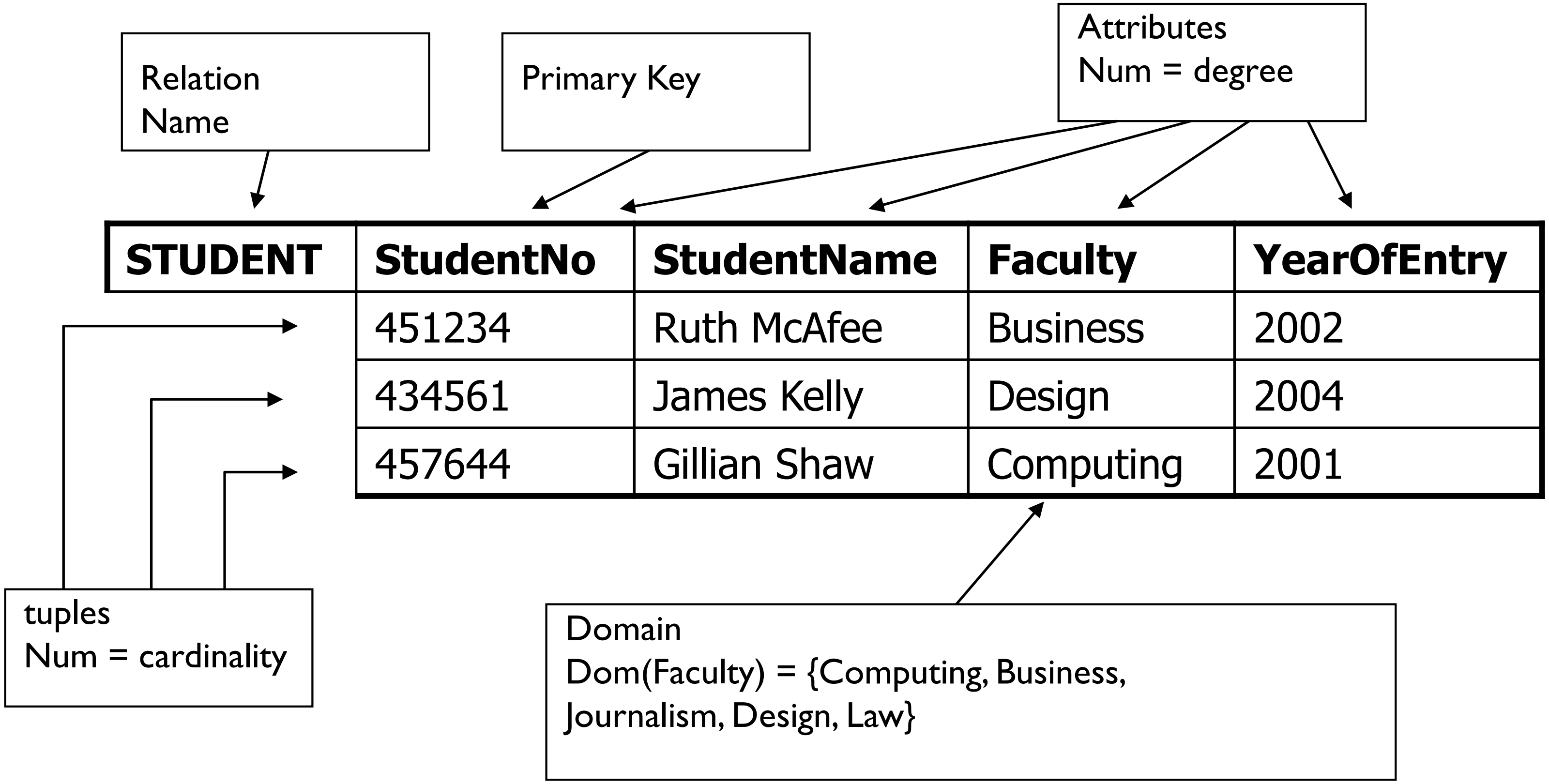
Relational Concepts

- Data is represented as collections of **relations**
- Each relation is **table** of values
- Each table consists of **rows** and **columns**
- Each **row** represents an **entity** or **record**
- Rows are unordered
- No duplicate rows are allowed
- Each row has a **primary key** which uniquely identifies the record/entity
- Each column represents an **attribute**
- Table name and Column name are used to help interpret the values

Database Terminology

- **Relation** is a mathematical term for a **table**
- **Row** is called a **Tuple**
- **Column** is called an **Attribute**
- **Domain** is used to describe the types of values that can appear in a column
- **Degree** is the number of attributes
- **Atomic Value** – precisely one value at each row intersection
- **Cardinality** – the number of tuples/rows in a relation
- **Null Value** – Missing, not known or irrelevant data (not the same as zero or blank)

Student Table



Relation Name

Primary Key

Attributes
Num = degree

STUDENT	StudentNo	StudentName	Faculty	YearOfEntry
	451234	Ruth McAfee	Business	2002
	434561	James Kelly	Design	2004
	457644	Gillian Shaw	Computing	2001

tuples
Num = cardinality

Domain
Dom(Faculty) = {Computing, Business, Journalism, Design, Law}

Data Independence

- Two types of data independence
  - Physical
  - Logical
- Physical is the idea that applications that use the data should not have to worry about detail of how it is stored
- Data Independence allows database to grow, shrink, add attributes
- Applications deal with the DBMS which in turn deals with the Database
- Differing degrees of success