Lecture 12 – Introduction to Databases

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

Objectives

- To list different types of databases
- To understand relational databases
- To examine the process of designing a database
  - (Next week)

- Today's practical
  - Basic use of Access database

The Relational Database Model

- The most common model of data today
  - But not the only one…
- Older models:
  - Hierarchical, ISAM, CODASYL…
- Newer models
  - Object relational
  - Free form
- The relational model involves
  - Entities
  - Relationships

Entities

- There are “Entities”
  - Entities are things we want to know about
  - Entities can be grouped by their “type”
- Entities have “attributes”
  - Attributes are what we want to know about the things
  - They may be calculated values (“virtual”)
- Entities are stored in a “table”
  - One entity per row
  - One attribute per column

Database Keys

- There must be a way to uniquely identify each entity in a table
- This is known as the <u>primary key</u>
- It can be a single existing attribute (e.g. chassis no. for a table of car entities)
- It can be a system generated attribute (sometimes known as a “sysid”)
- It can be a combination of two or more attributes
  - E.g. To identify a specific flight sector we need
    - The flight number (ABC123)
    - The origin (LHR) – because flights can be mutli-sector
    - And the date of the first sector (consider delayed flights)

Indexes

- The primary key is normally also indexed
- Other attributes may also be indexed
- Indexes speed up searches based on this attribute
- But they are not “free”
  - Indexes take up disk space
  - They can slow insertions, deletions and modifications of data
    - Because the index has to be updated
- Indexes can usually be added later if required
- Part of the “tuning” process

Relationships

- Relationships are “links” between entities
  - E.g. “Is written by”, “Is a kind of”
- Relationships can have “constraints”
  - One-to-one, one-to-many, one-to-four
- Relationships can be recursive
  - An assembly consists of one or more components, <u>or other assemblies</u>

Referential Integrity

- Ideally, the database will enforce referential integrity
- I.e. it will ensure that all relationships are valid
- For example, consider a table of “vehicles”
  - Has attributes including “reg_no”, “colour”, …
- And has relationship with the table “vehicle_types”
  - Contains rows for “car”, “van”, “mpv”, “minibus” etc.
- If “vehicles” contains a with a given vehicle type
  - E.g. “V91 DJB”, “burgundy”, “mpv”, …
- The database will NOT allow the row “mpv” to be deleted from the “vehicle_types” table
  - Must delete all related “vehicles” first

Forms

- The most common way to view, update and enter data
- Allows viewing, entry or update of a single entity at a time
- Some tools can generate forms automatically
- More usefully, developers can create their own
  - Very similar to GUI builders in IDEs
- May include Data Validation
  - Ensuring range, format etc. of data
  - (Not the same as referential integrity)

Views

- Caution! This is an overloaded term!
- In Access a “view” is a way of organising the data on the screen
  - Forms view - (as above)
  - Grid (or table) view – Like a spreadsheet
- In other systems a “view” is a “virtual table”
  - Allows different users to have data organised to their needs
  - But involves no duplication of data

Triggers and Stored Procedures

- Some databases allow the developer to store "code" in the database
  - Java, C++, SQL …
- The code can be called when particular "events" happen
  - Typically on Insert, delete, update
- Can be used to:
  - Cause external activities to be started
  - Build synchronisation code
  - Enforce "Business Rules"
    - E.g. Not more than 3 cars & 2 mpvs or 2 cars & 2 vans to be booked for service on the same day

Structured Query Language

- Known as SQL (sometimes pronounced “sequel”)
- Now an international standard
- Allows queries to be constructed
  - Select <attributes> From <table> Where <conditions>
- Provides database maintenance facilities
  - Create & drop tables, indices and views
- Does NOT have procedural elements
  - Variables, control structures, loops etc.
- But can be “embedded” in other languages
  - Visual Basic, PRO*C

COTS Database Packages

- At first only “flat file” databases
  - A single table of similar data (e.g. dBase)
- Later, more sophisticated databases
  - Multi-table databases (e.g dBase II, MS Access)
- Also, mini and mainframe products migrate to PC
  - PC Oracle
- In future practicals we will look at Microsoft Access
  - But simple flat file databases only
- Next year we will cover advanced Access topics
  - Multi-table databases, programming
- And look at Oracle, the market leading database

Next Week

- We will look at Data Modelling
  - How should we structure our database?
  - What entities, relationships and attributes do we need?
  - What are the primary keys? Indexes?
  - What business rules do we need?
- And issues of database scalability
  - Is there a difference between PC and mainframe / minicomputer databases?

Today’s Practical

- Basic use of Access
- There is a lot of material to cover – you may NOT be familiar with all of it

- <u>REMEMBER TO SIGN OFF ON THE REGISTRATION SHEET!</u>