MSc in Official Statistics
Statistical Computing:
Data Structures and Objects

Andrew Westlake

Survey & Statistical Computing

63 Ridge Road, London N8 9NP, UK

+44 (0) 20 8374 4723

AJW@SaSC.co.uk (E-Mail)

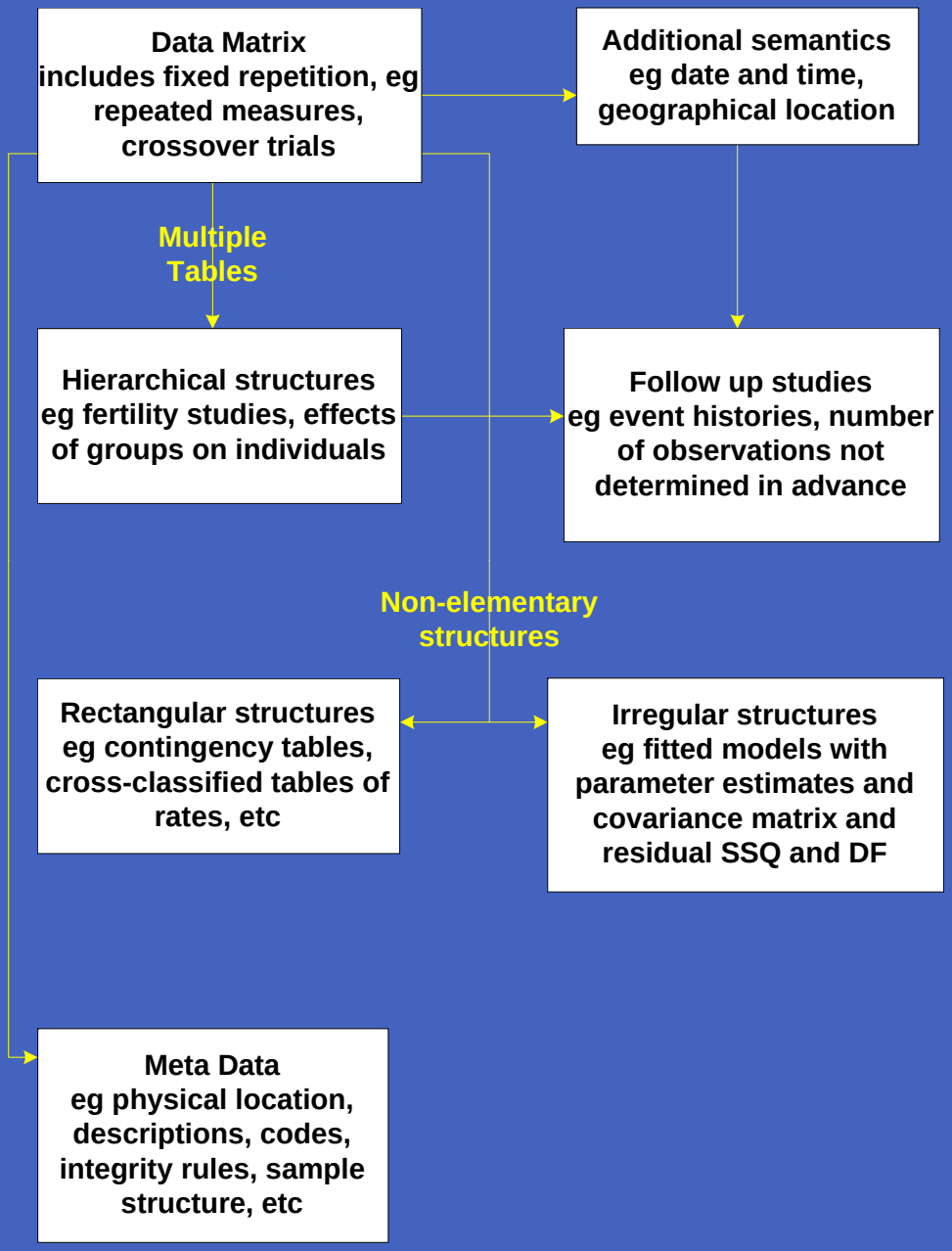
www.SaSC.co.uk

Overview

- Computing
  - » DataBase ideas are important (see Date)
  - » RDBMS are the main type available (see Codd)
  - » Good implementations exist
  - » SQL is an important standard (see Date & Darwin)
  - » Object-oriented ideas are pervasive in programming
- Abstraction is important
  - » Get the right structure, simplify maintenance of the solution
  - » Follow appropriate standards (many types and levels)
  - » Provides conceptual framework, simplifies communication
- Statistics
  - » Statistical problems are different
  - » Important to take a broad, well-informed view
  - » Ideal solutions are elusive

Data Structures in Statistics

- RDBMS handles basic data matrix well
- Other data models needed for more complex structures
- Object approach has more flexibility, but functionality for data manipulation has to be programmed



Importance of structure in data

- Micro, macro, meta data, models, results, conclusions
- Formalisation => automation
- Generalisation => generic facilities
- Conceptual framework aids thinking and communication
  - » Conceptual frameworks provide a focussed language for communication
  - » Standardised functionality, tailored through parameters, is easier to understand than code in a general-purpose language
- Need for semantics in addition to structure
  - » Middle ground of standard structures (object classes) with generic behaviour
    - Cf. RDBMS and SQL

Relational Concepts & Systems

- Very useful way to think about data
- Useful context for direct manipulation of data
- Provides framework for using data in processing systems
- Does not address design of processing functionality
- Object-Oriented approach provides a conceptual framework for systems

The Object Paradigm (1)

- An *object* is a structured collection of information
  - » An *instance* of a particular type of component
  - » Examples might be classification, variable, label, dataset, summary table, …
- The general definition of a particular type of object is called a *class*
  - » Not a particularly good choice of name
  - » A class has behaviour (methods) as well as structure
- The specification of a class determines the structure and semantics of the objects that are instances of that class
  - » The objects can contain different information, since they describe different instances, but their structure and behaviour is the same

The Object Paradigm (2)

- The specification of a class includes the *attributes* which form its structure
  - » Can be simple (such as numbers or strings)
  - » Or complex (effectively links to and collections of other objects)
- Every object (instance) has a unique *identity*
  - » This can be referenced by other objects
  - » Object identities are *global*, so object references do not need different forms for different types of object

Object-oriented Concepts (1)

- Identified objects
  - » Every object, whatever its type, has a unique and identifiable existence
  - » Can ask an object about its type, name, etc.
  - » Contrast with relational sets, which are based only on data values
    - In relational model, cannot have two tuples (rows) with the same set of data values (so introduce ID values)
- Classes, methods and properties
  - » Every object is an instance of a particular Class
    - Has properties (attributes) and methods (the operations it can perform)
    - Can be public and private
  - » Can only access an object through its public interface (properties and methods)
    - This is to prevent side-effects
  - » Properties can be complex structures, including collections of other objects

Object-oriented Concepts (2)

- Inheritance
  - » A class can be defined as a specialisation of another, and inherits all the definitions of the parent
- Extension
  - » Can add additional properties and methods to the child class
- Polymorphism
  - » Can alter the definition of a property or method within the child (but not the pattern)
    - E.g. change the Print() method to include additional properties
  - » Invoking the method uses whichever version is appropriate to the particular object
    - Object.Print

Object-oriented Concepts (3)

- Associations between Classes
  - » Relate to relationships between objects
  - » Many types
    - E.g. Person A 'is married to' Person B
    - Summary table uses dataset and classifications
    - HIV record extends basic Patient record
    - Variable has value set and validation rules and question text and scope
- Levels of Abstraction
  - » The classes in one model can be instances of the classes of a higher level model (sometimes called a metamodel)
  - » What is abstract depends on context
    - A particular Variable is an instance in the Metadata context, but a class in the data context

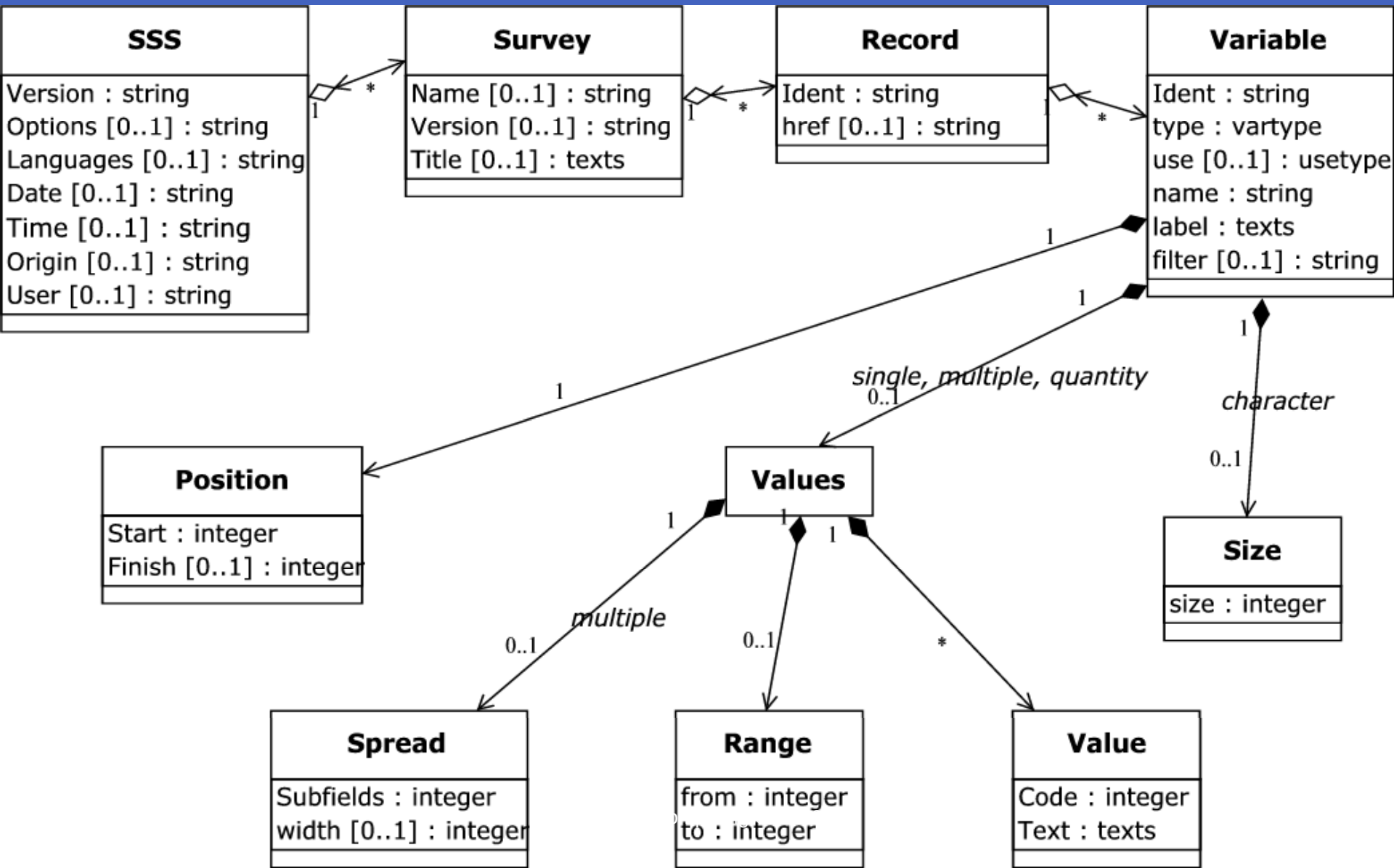
Object-Oriented systems

- Most modern programming languages
  - » C++, Java
  - » Visual Basic (strictly, it is object-based)
  - » S, R
- Some specialised Database systems
  - » ODABA, ObjectDB, etc.
- No mainstream DBMS
  - » Some (e.g. Oracle, SQL Server, Postgres) have object extensions
    - Compound values in tables
  - » Object extensions in SQL3

Scope of O-O ideas

- Can apply object concepts to non-object systems
  - » E.g. in RDBMS, think of the concept of a table as a class, with actual table definitions as instances
  - » In turn, a table contains the class definition for its rows, and the actual rows are instances
- Can use to think about any structure and process
  - » Not restricted to computer systems
- Standard diagram and design systems (UML)
  - » Useful for expressing any ideas
  - » Constructs are precise, if used correctly
  - » Includes Semantics as well as structure

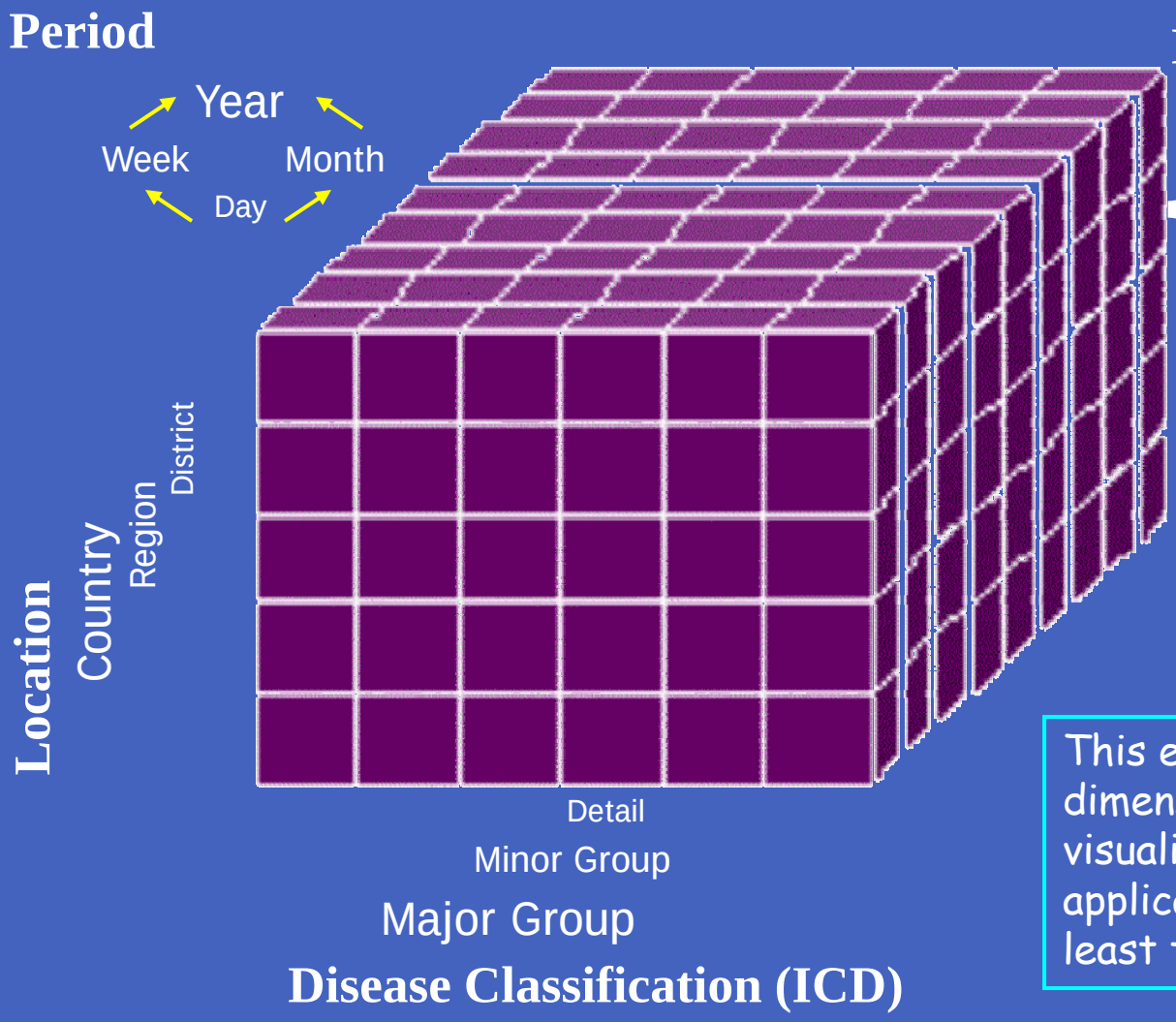
Triple-S structure modelled in UML



Statistical Summaries

- Various forms
  - » Statistical Analysis, with fitted model
  - » Statistical Aggregates, presented on paper
  - » Statistical Diagrams
- Often confuse presentation with structure

Aggregated Results, as Multi-way Table



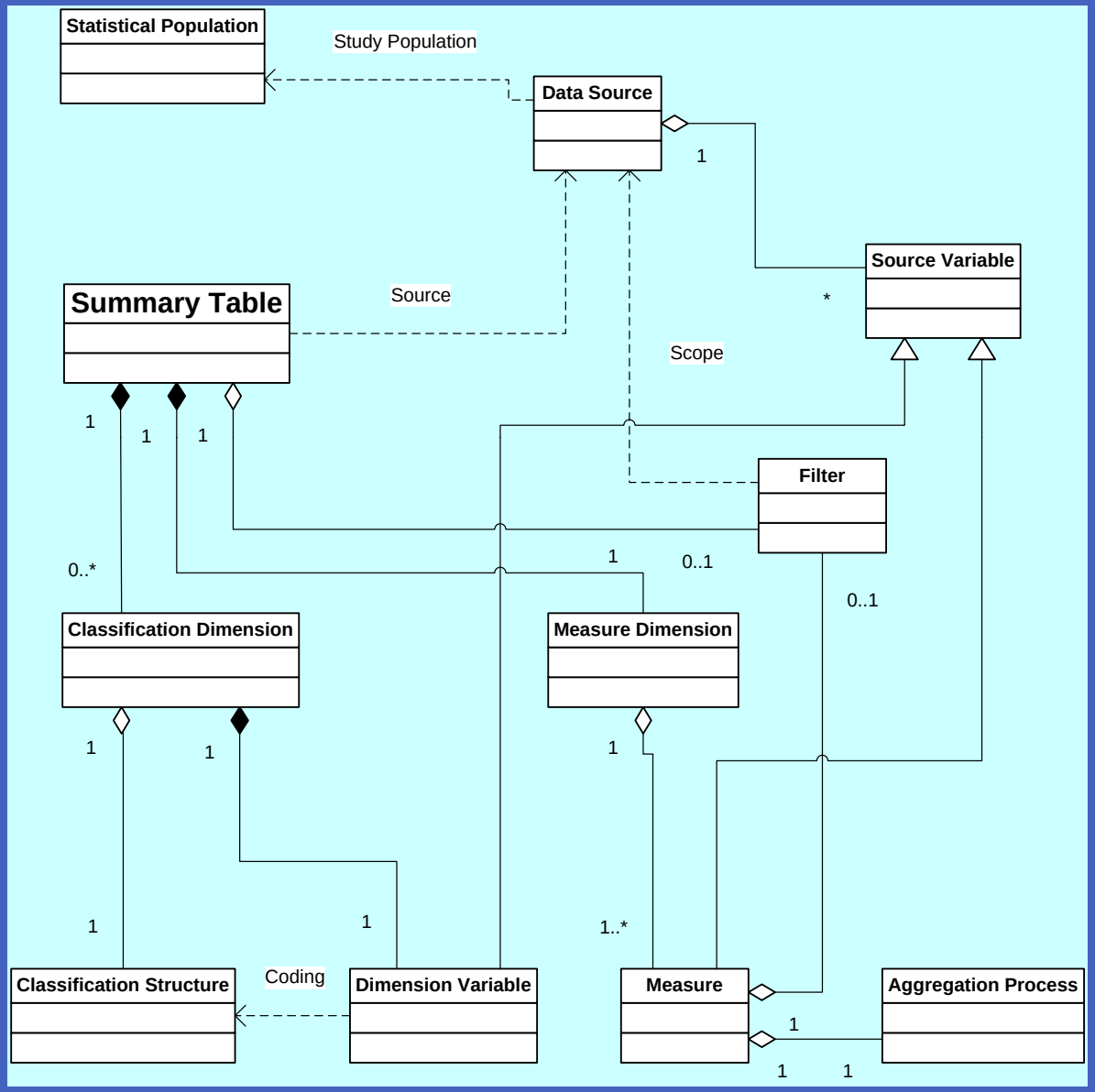
This example has three dimensions (so that it can be visualised). In reality, for this application, we would need at least two more, Age and Gender.

Aggregate Data Components

- Statistical Table
  - » Multidimensional (Cube) structure of Classification dimensions
  - » Cells contain Measures (really an additional dimension set)
  - » Defined with respect to a Statistical Population and data source
  - » May have filters to restrict scope
- Classification Dimension
  - » Uses a Classification Structure
  - » Linked to a source variable that is coded according to some level of the classification
- Measure Dimension
  - » Each category is a different measure
  - » Each measure has source variables and an aggregation rule, which may include a filter

Statistical Summary in UML

- A possible structure for Statistical Summaries
- Correct structure is necessary, but not sufficient
  - » Need Functionality and Semantics as well



Manipulation Functionality

- Store information with minimal aggregation
  - » Maximum detail in classifications
  - » Further aggregation (to less detail) on demand
    (may pre-compute for efficiency, may retain original records)
- Algebra for aggregation of classifications and measures is basically straight forward
- Aggregation of Measures (less detail)
  - » Everything based on summation can be regrouped
    (cf. updating algorithms, sufficient statistics)
  - » Some others, e.g Range
  - » Special issues for time: aggregate or cross sectional measures
- Derivations, across measures, cells, classifications, tables
- All aggregated tables are **proper** tables

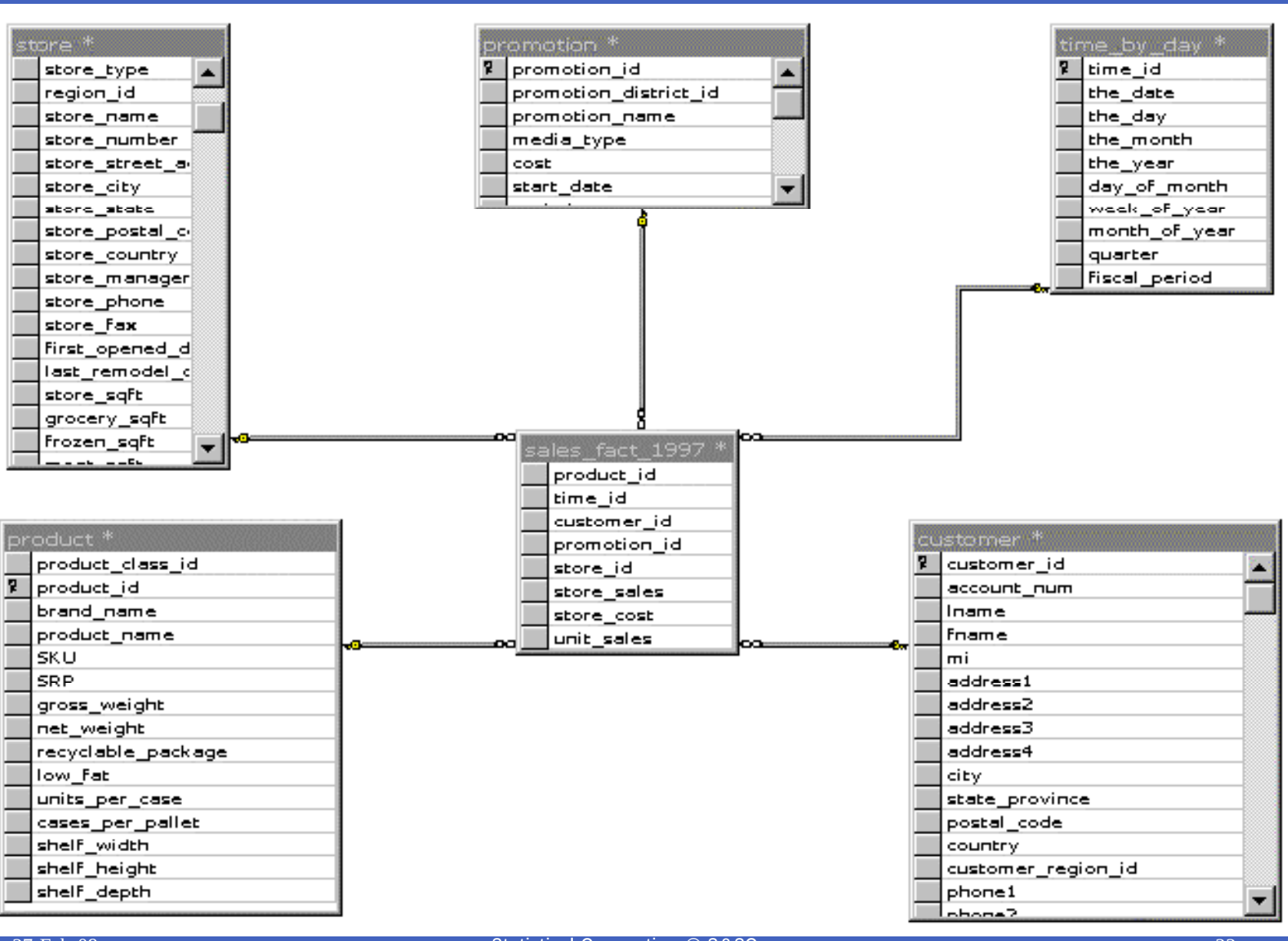
Presentation Functionality

- Mapping from logical structure to presentation layout
  - » Rows, columns, pages, margins
- Combination of separate tables
  - » Concatenation within conformable dimensions
  - » E.g. Smoking and Drinking rates by Age
- Presentation tables do not need to be proper tables
- Dynamic functionality for on-line presentation
  - » Layout, roll-up, drill-down, derive new
  - » C.f. Neighbourhood Statistics

Data warehouses and OLAP

- Data Warehouse
  - » Database of information extracted from other operational systems (so relatively static)
  - » Large volumes, so makes use of special physical optimisation
  - » Objectives – Business intelligence, statistics(?)
- OLAP
  - » On-line Analytical Programming (term invented by Codd)
  - » Extension of cross tabulation
  - » Dynamic exploration, subset identification (data mining), not modelling
  - » Potentially useful for statistics, but needs extension
    - Manipulative and publication functionality
    - Limited awareness of data semantics and metadata

Star Schema


store *
store_type
region_id
store_name
store_number
store_street_a
store_city
store_state
store_postal_c
store_country
store_manager
store_phone
store_Fax
first_opened_d
last_remodel_c
store_sqft
grocery_sqft
frozen_sqft
promotion *
promotion_id
promotion_district_id
promotion_name
media_type
cost
start_date
time_by_day *
time_id
the_date
the_day
the_month
the_year
day_of_month
week_of_year
month_of_year
quarter
fiscal_period
sales_fact_1997 *
product_id
time_id
customer_id
promotion_id
store_id
store_sales
store_cost
unit_sales
product *
product_class_id
product_id
brand_name
product_name
SKU
SRP
gross_weight
net_weight
recyclable_package
low_fat
units_per_case
cases_per_pallet
shelf_width
shelf_height
shelf_depth
customer *
customer_id
account_num
lname
fname
mi
address1
address2
address3
address4
city
state_province
postal_code
country
customer_region_id
phone1

Summary

- Rich set of structures needed for statistical information
- Often have to compromise
  - » Better to do this from a position of understanding
- Object concepts useful and widely used by computing specialists
- RDBMS systems are very useful for statistical data