**Database Systems**

**General Knowledge**

**ACID -** *Transactions are tools to achieve the ACID properties.*

* **Atomicity**

Can guarantee that all of a transaction happens, or none of it does

* **Consistency**

Data will be consistent; none of the constraints you have on related data will ever be violated.

* **Isolation**

One transaction cannot read data from another transaction that is not yet completed. **Mutual Exclusion**

* **Durability**

Once a transaction is complete, it is guaranteed that all changes have been recorded to a medium (e.g. hard disk), and transaction is recorded as complete.

**What is a Database?**

A managed data store- NOT an information store

* Information System is an application that uses a database
* **Information** is **Data** in **Context**

**Universe of Discourse - Order Created and preserved**

* **Discourse** is the **story** that **data can tell**
* **Universe** is the **boundary** **of** that **story**

*DB Data is about a specific topic, not random set of data items.*

* **Data** can and must be **structured**: **build a Model -** *Schema*
* So the data can and must be **managed**: **Integrity**

**Database Management System**

*Components of a (DBMS) include:*

**Database Engine**- *Does the Work –* We See Tables, Not Data

* Maintains and accesses data
* Concurrency control
* Security
* Integrity

**Data Dictionary** – *DB within DB* - Holds information about:

* names & location of tables
* definitions of tables (schema)
* view definitions
* index information (speed up retrieval of data)
* access rights

**User Interface –** Two Types – *End user* & *system developer*

**System development**

* Schema manager & Query processor
* Forms, Report & Menu generator

**Procedural Language Facility**

**Data Management**

* **Holds data**: facts, not errors
* **Persists**
* **Manages itself** (intensional) and its data (extensional)
* **Preserves** **the** **integrity** of the data

Without redundancy

Handling transactions: changing the state of the database

* **Static constraint**: *doesn’t allow invalid data*
* **Transition constraints**: *domain rules not broken*

**Database Functions – Preformed on the data**

* Update functions: change the data (within the limits of the integrity constraints)
* Query functions: check if the database is holding certain facts, and how are these facts qualified

**Formalism**

How Data is modelled in specific DB model

* A RDB models entities as freely related to other entities by common properties
* An object database models classes of data in hierarchical arrangements associated to other hierarchies

**Multi-User/Multi-Access**

* Manages secure simultaneous access by multiple users in different places
* Provides the ability for different users to have selected access into different parts of the set of facts.
* Different parts of the set of facts can be distributed into various physical stores

**Other Types of Databases**

**Database models and Techniques**

**Types of Database**

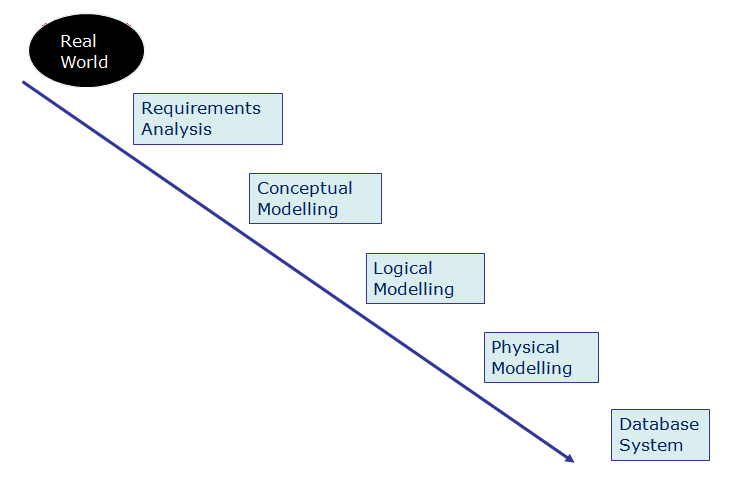
* Relational database
* Object database model
* Data Processing Technique - Data mining

**Selecting a Database**

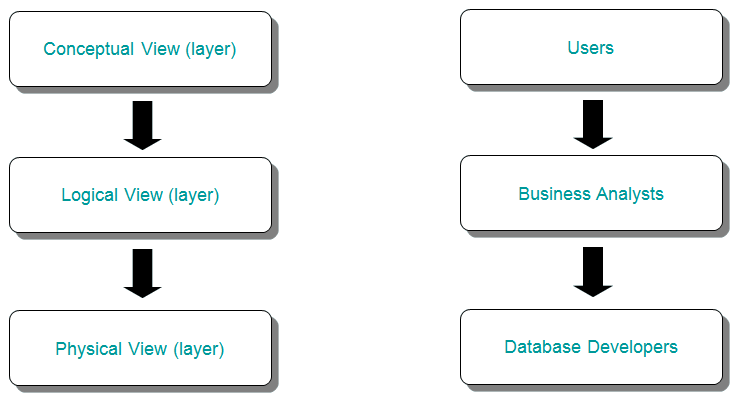
* Size and complexity of data set
* Data Types Involved
* Volatility of Data
* Nature of Transactions
* Nature of queries that may be performed on data

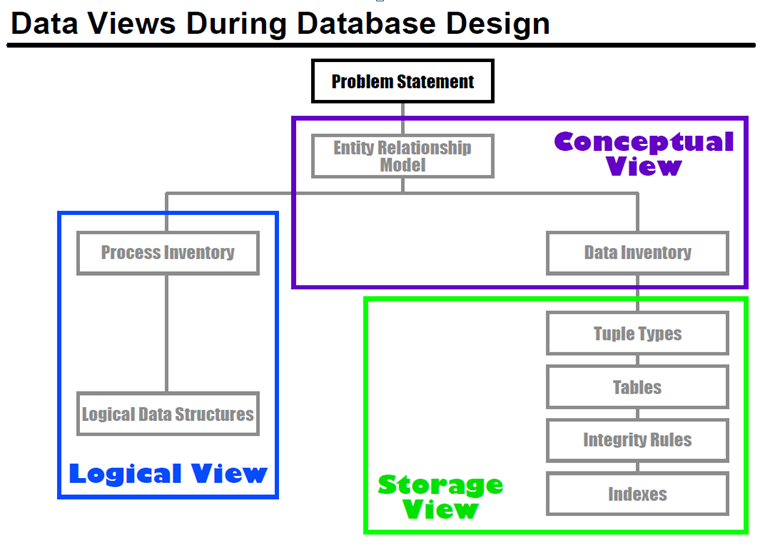
**Relational Databases**

**Database Design**



The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems” Information Systems - 1978





**Requirements Elicitation and Analysis**

* Identify the **stakeholders**
* Identify the **structure** and **use of data** by the stakeholders

**Research Methods** – *Interviews, Surveys, Observation, Examine Documents, Prototype*

***Requirements Analysis***

* What Clients want from a DB supported application

**Reports**

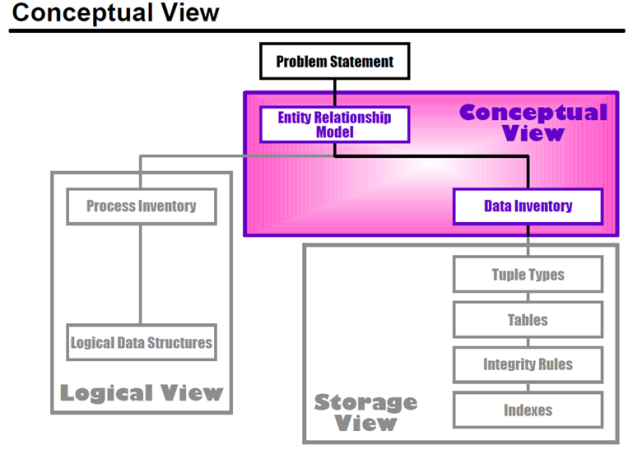
* What questions do clients want to ask the DB

**Queries** use **Forms** to **ask** & **display** query results

* What data Is needed to answer queries

**Forms** needed to enter data

**Conceptual Model**



*Describe the users’ models of the database in a way that can be* ***agreed and understood by both users and developers***

* *Usually* ***described via an E-R diagram***

*Also known as the User model*

* + How the users think about the database from the perspective of the application domain
  + Process- and implementation-independent

**Entity- Relationship Diagram**

Entity-Relationship diagrams describe the users’ models.

* + **Entities**
  + **Relationships**
  + **Attributes**

**Approach to conceptual Model**

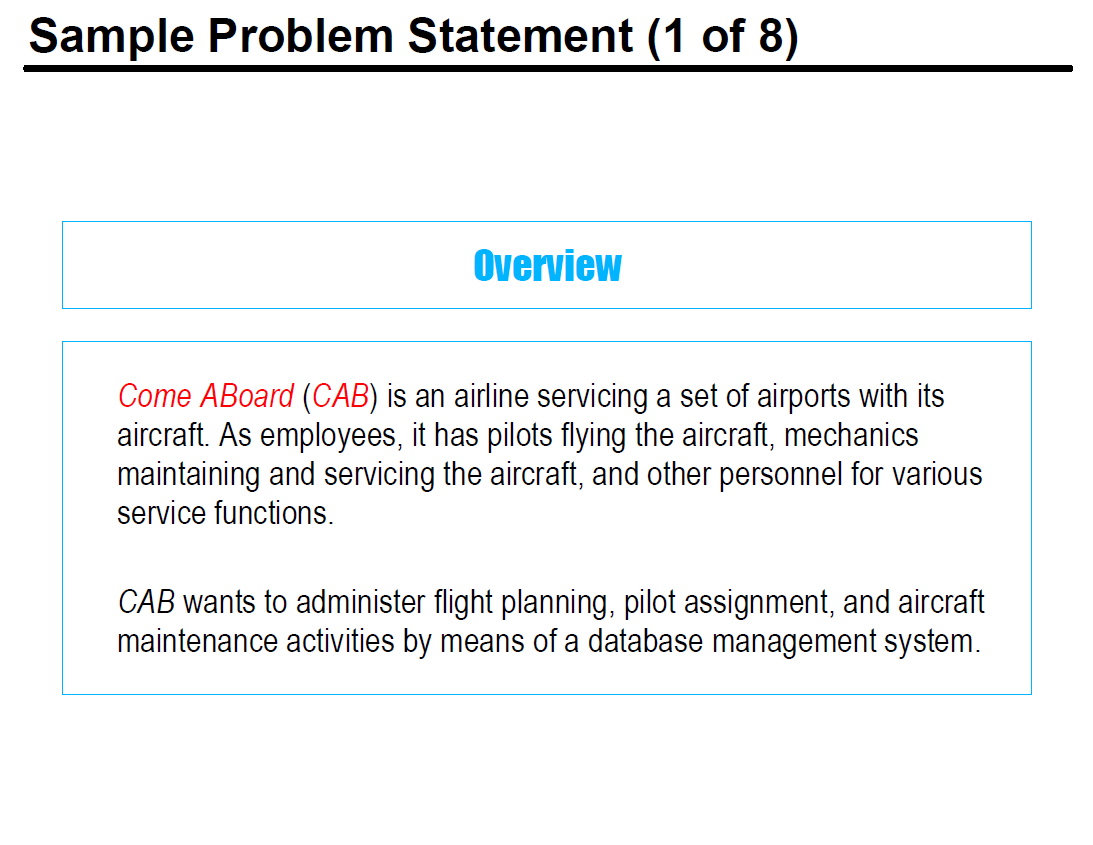
1. Identify **entity types**
2. Identify **relationship types**
3. Identify **attributes with each of above**
4. Determine **attribute domains**
5. Determine **keys**
6. **Validate model**
7. **Review model with user**

**Problem Statement**

* **Created** by *application domain expert and used to* ***understand Application Domain***
* **No detailed information**
* **Create** **entity-relationship model** for the application domain

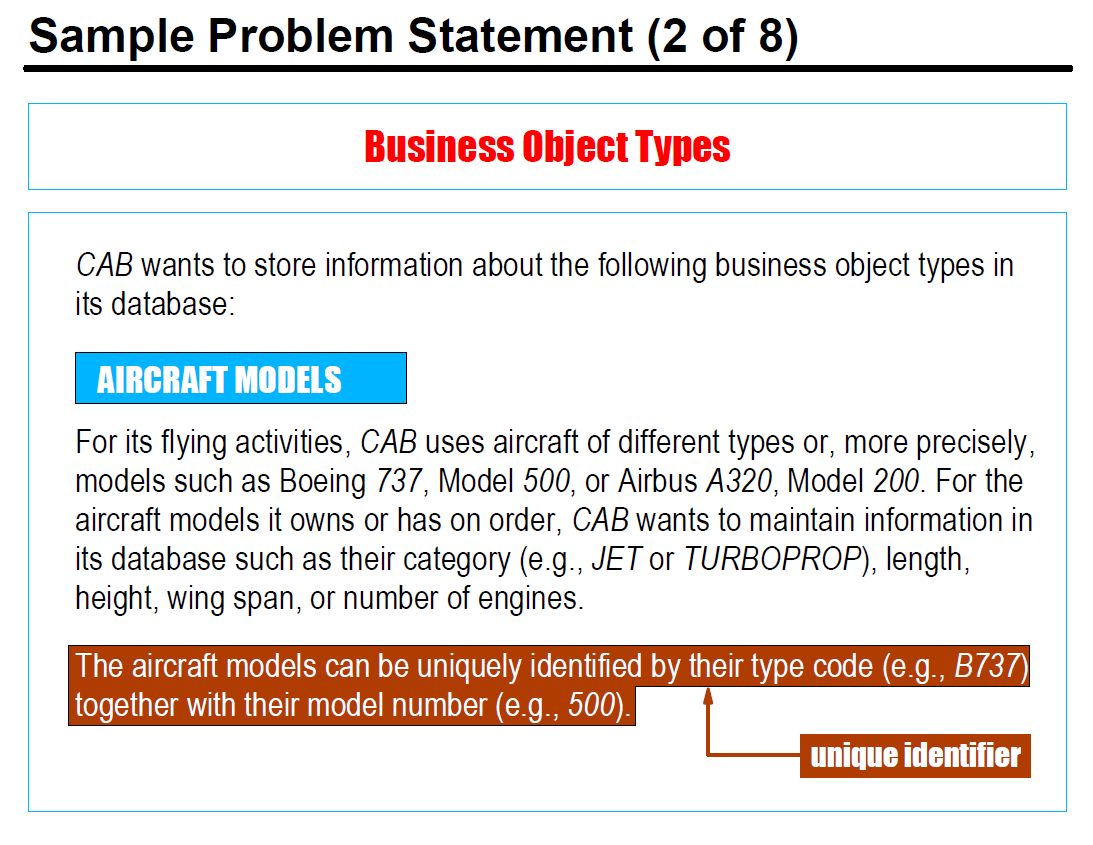
**Overview** - ***Short Textural description***

**Overview of Application Domain** - What it does and what is to be achieved by the DBMS.



**Business Object Types** - A listing of all **business object types**

**Textual description** - How individual objects of the business object type can be identified



**Confirming entity types**

* Can entity instances be uniquely identified?
* Do the entity instances also have non-key attributes?
* Will there definitely be multiple instances of that type for the application domain?
* Is there a generic class name?
* Are the instances meaningful by themselves? (rather than being meaningful only when processed with instances of another entity type)

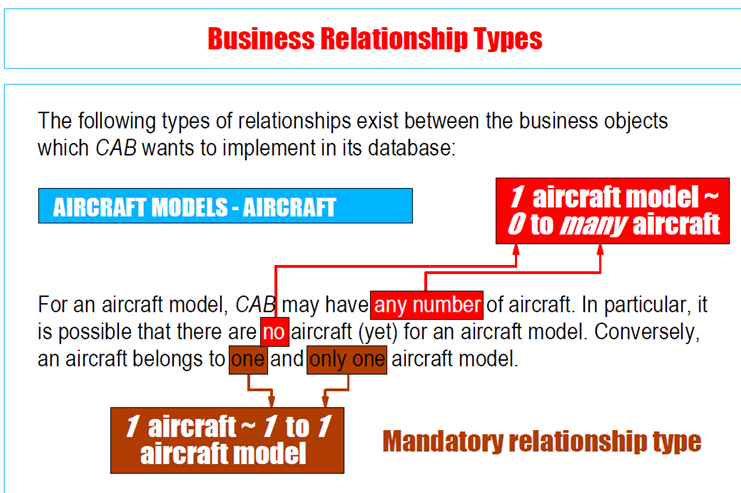
**Business Relationship Types**

A listing of all types of **relationships** **between the business objects**

* + **Each** of the **type of relationship - has** *Textual description*
  + How many **relationships** of the same type an **object can have**

*Must there be at least one?*

* + If one object is *deleted*, what about the other(s)? **Constraints**

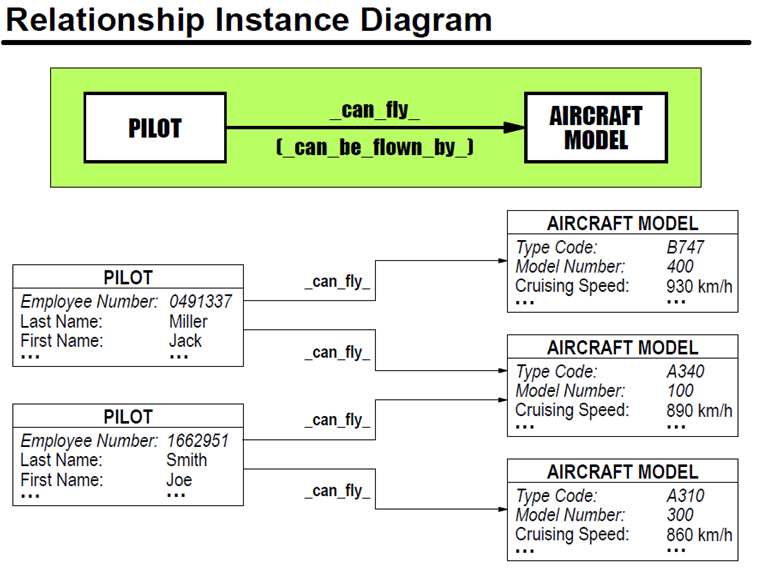


**Confirming Relationship Types**

* Describes the association between
  + entity instances (not necessarily of different types) or
  + relationship instances or
  + an entity instance and a relationship instance
* Relationship types are typically binary (interconnect two entity instances)
* Entities can be related by a relationship label
  + Cheese *is an ingredient of* Lasagne
  + Lasagne *is a dish* in a Meal
* Relationships that are *required*
  + (not relationships that are only *possible*)
* Relationship types can be viewed from two directions (primary, inverse)

**Instance Diagram – Confirm Relationship Types**

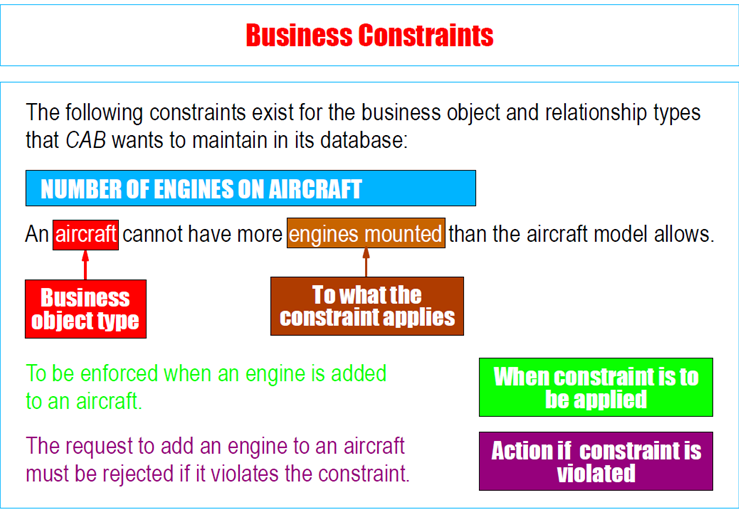
Entity instances are interconnected to better visualise



**Business Constraints**

A listing of all types of **all constraints for the business object types and/or business relationship types**

* + Textual description of the constraint
  + The object and relationship types to which the constraint applies
  + When the constraint is to be applied
  + Action to be performed if the constraint is violated



**Logical Modelling and Normalisation**

a

**Physical Model**

**Relational Databases**

**Fundamentals**

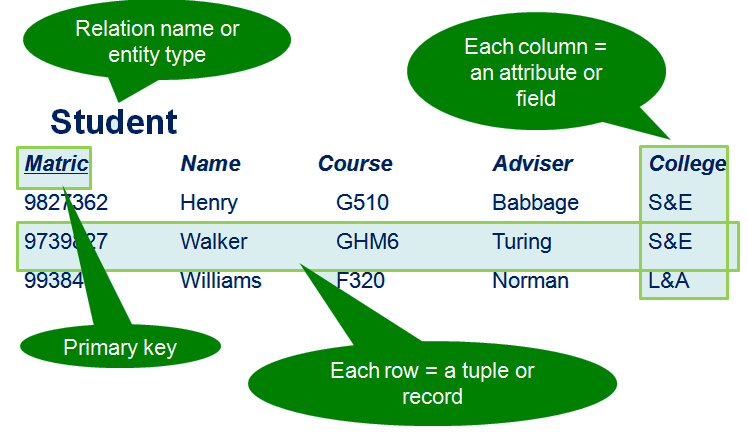
Data model based on **relations**, ie **tables** with **columns** and **rows**.

* **Data** is **broken up into relations** - *To help the data processing*
* **Set** of **data** held in a **relation**
* *Relation = Table = Entity Type*
* **Application domain (Database)** is consists of table collection
* **Relationships** are established between tables based on **Keys**

Entity Type = Thing / Object e.g. Fruit

Entity instance = Instance of that thing/Object e.g. Apple

**Relation – Summary**



* **Row** is Tuple
* **Table** is Relation
* **Intension** is the Relation Column Title Headers
* **Extension** is adding records to the relation
* **Cardinality** is the number of Tuples in Relation

**Rules of Relational theory**

1. Each value in a **relation is atomic**

*For each row, the value within a column is always one single value and never a group of values*



1. The values of an **attribute are all of the same kind.**

*Those defined on the* ***domain*** *for that attribute*



1. Columns (attribute) have **distinct identifying name**



1. **Each row** (tuple) is unique - each relation **has** **(unique) ID**



1. There is no significance in the ordering of rows and columns
   * you can only refer to a column by its name, not its position
   * you can only refer to a row by its primary key, not its position



**Attributes, Keys, & Constraints**

**Attributes** represent properties of an entity type.

* Each instance of an entity type has its own particular values for attributes.
* Attribute values make up the main part of the data in a database.

**Identifying attributes**

* **Elementary attributes:** street, city, postcode
* **Composite attribute:** address
* **Single-valued:** primary abode
* **Multi-valued:** email address
* **Derived attribute:** date of birth

**Attribute domain -** A domain is the valid set of values that an attribute can hold. - **Dish:** CHARACTER(30)

**Identifying attribute domains**

* **Allowable data types -** eg integer
  + **Allowable sizes -** eg number of characters
  + **Allowable formats -** eg ../…/…. for date
* **Allowable set of values**

eg 0<..<32, in the set {Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec} for …, 1900<….<2012

**Keys**

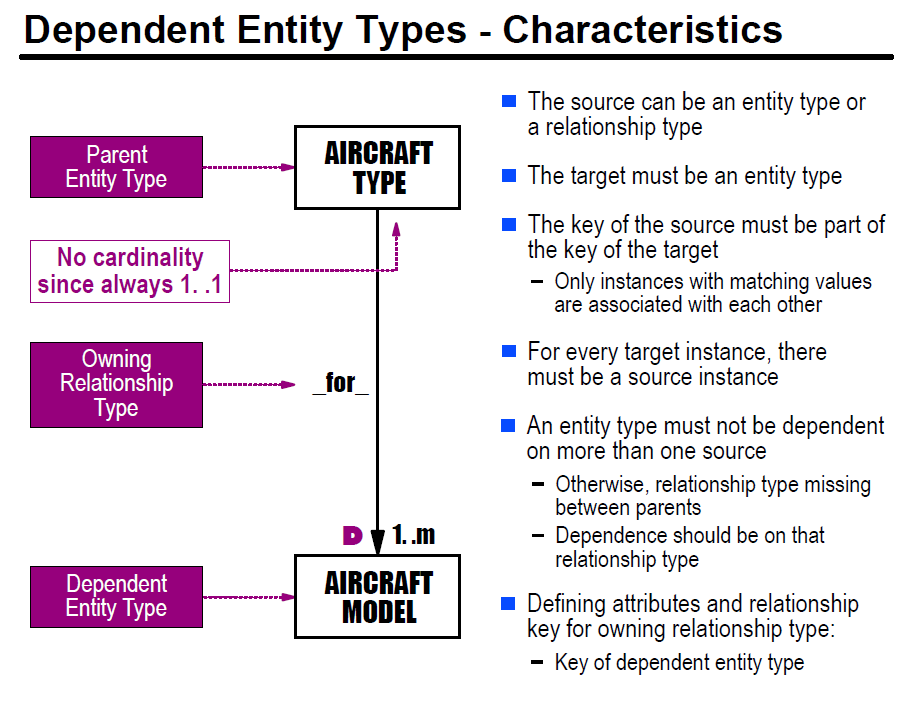
**Candidate keys -**Minimal set of attributes that uniquely identifies an occurrence of an entity

**Primary Key** attributes uniquely identify the entities within an entity type - *One of the candidate keys*

* With the least number of attributes
* Least likely to change and easiest to use
* With the fewest characters or smallest maximum value

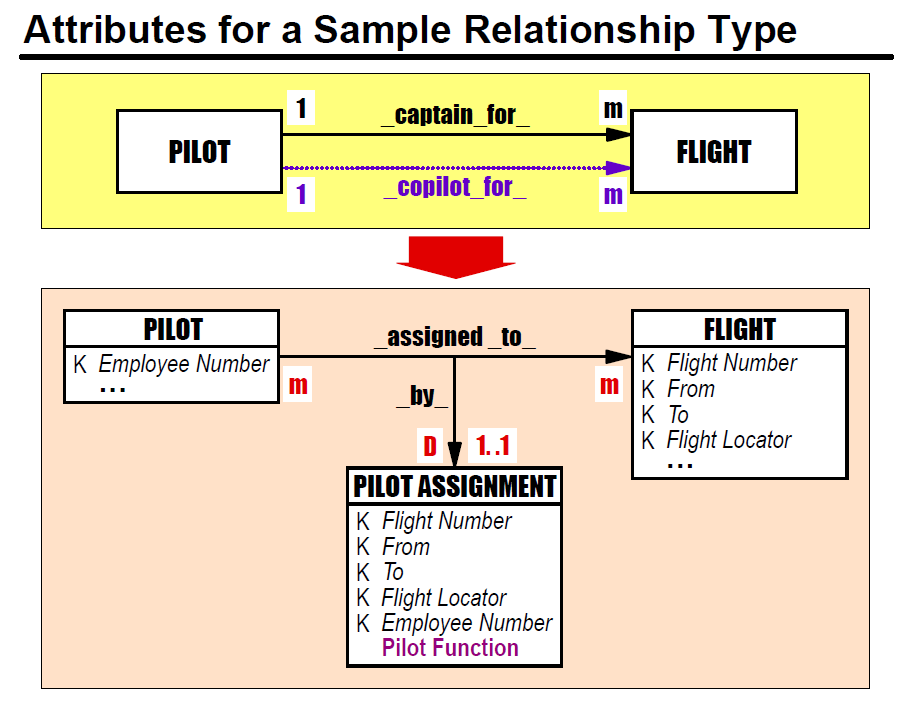
**Foreign Key** attributes link an entity within one entity type to the entities within another entity type with which it has a relationship. **-** All the other candidate keys

* The primary key is placed in the other relation to indicate a relationship between the two relations

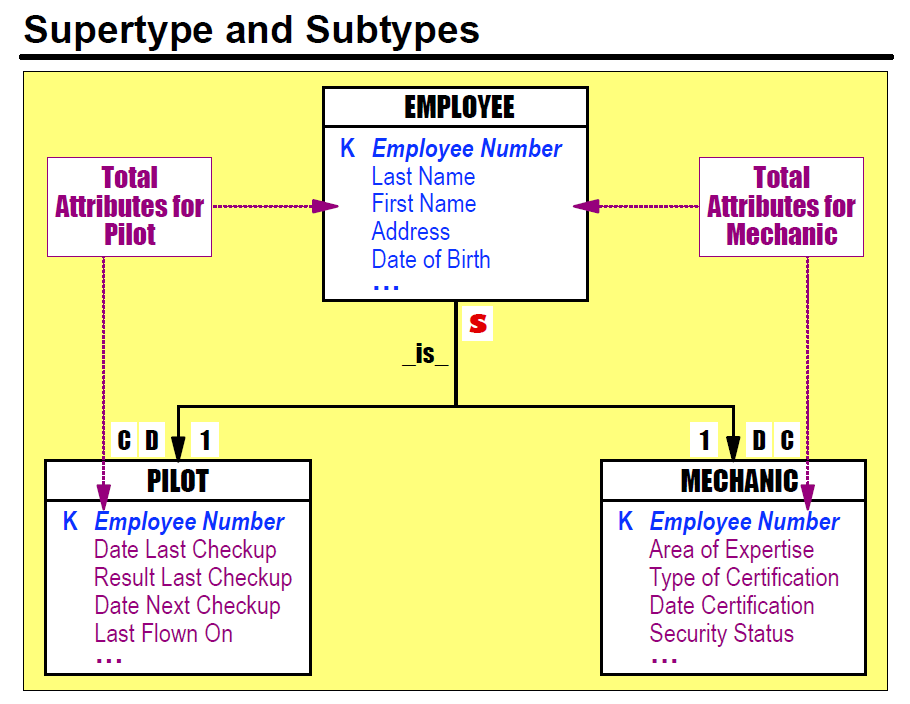


**Modeling relationship attributes - Model 1 - Relationship**

Dependent entity types can help model attributes of a relationship



**Modeling relationship attributes - Model 2 – Super/Subtype**

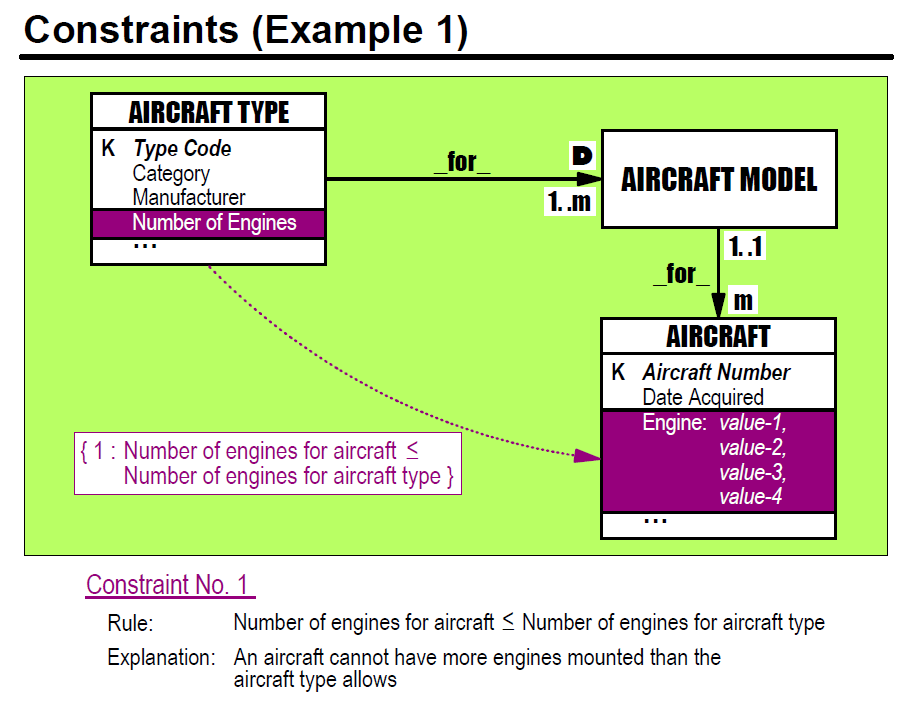


**Validate the model**

* Identify the actions or set of actions carried out by a user that access or change the content of the database.
* This action/these actions represent a transaction.
* Attempt to perform the transactions manually using the model.
* Transactions represent real events that must be processed via the DBMS.
* The database integrity must be maintained when a transaction is performed.
* To identify transactions, think of data needed, characteristics of the transaction and the resulting outputs.
* Mostly retrieval or update or combination.

**Constraints**

* Constraints restrict possible instances of entity or relationship types.
* Constraints have:
  + Constraining objects
  + Constrained objects
  + A rule
* Usually these come from business constraints.
* Show as braces around identifier and rule:
* {identifier : rule}
* If many objects, then use dotted line and write constraint near line.



**Transforming E-R Model to Data Dictionary Descriptions: Schema**

**Define each entity**

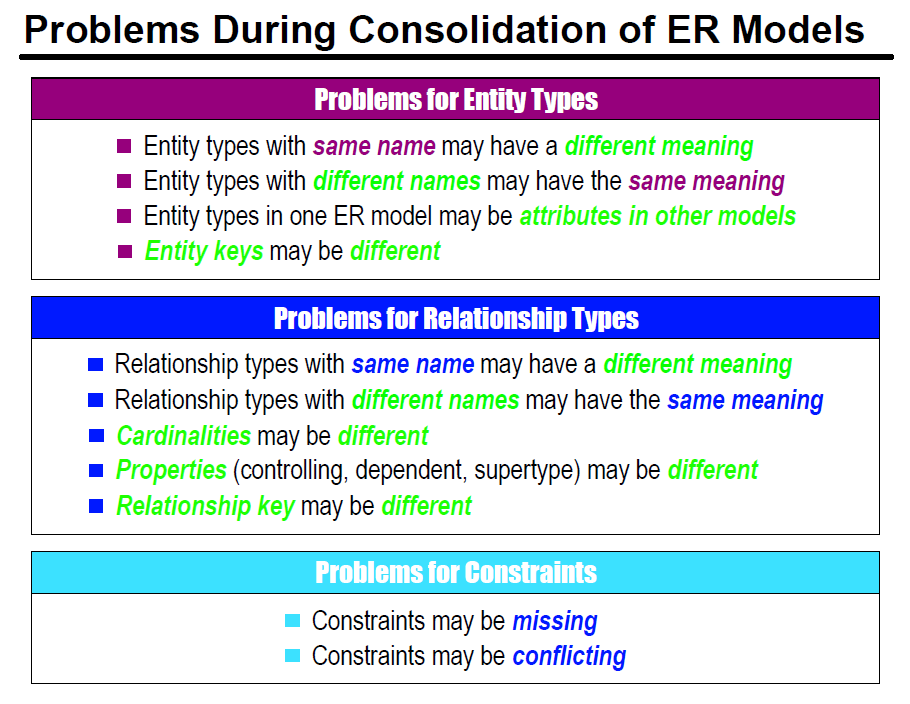
* + Entity Name:
  + Description:
  + Identifying Key Attribute(s):
  + Other Attributes:

**Define each relationship**

* + Relationship ID:
  + Relationship Name:
  + Description:
  + Participating Entity (xN):
    - Cardinality
    - Optionality

**Define each attribute**

* + Attribute Name:
  + Description:
  + Type:

****

**Relational Schema**

**SQL**

**SQL**

**DDL**  
**Data Definition Language** (DDL) statements are used to define the database structure or schema. Some examples:

* CREATE - to create objects in the database
* ALTER - alters the structure of the database
* DROP - delete objects from the database
* TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
* COMMENT - add comments to the data dictionary
* RENAME - rename an object

**DML**  
**Data Manipulation Language** (DML) statements are used for managing data within schema objects. Some examples:

* SELECT - retrieve data from the a database
* INSERT - insert data into a table
* UPDATE - updates existing data within a table
* DELETE - deletes all records from a table, the space for the records remain
* MERGE - UPSERT operation (insert or update)
* CALL - call a PL/SQL or Java subprogram
* EXPLAIN PLAN - explain access path to data
* LOCK TABLE - control concurrency

**DCL**  
**Data Control Language** (DCL) statements. Some examples:

* GRANT - gives user's access privileges to database
* REVOKE - withdraw access privileges given with the GRANT command

**TCL**  
**Transaction Control** (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

* COMMIT - save work done
* SAVEPOINT - identify a point in a transaction to which you can later roll back
* ROLLBACK - restore database to original since the last COMMIT
* SET TRANSACTION - Change transaction options like isolation level and what rollback segment to use

**MySQL VS SQL**

**Schema Manager**

Definition of structure of database - SQL

CREATE TABLE Committee

(

CommCode NUMBER(3) NOT NULL,

CommName CHAR(20),

CommChair CHAR(20)

)

**Large Database Systems**

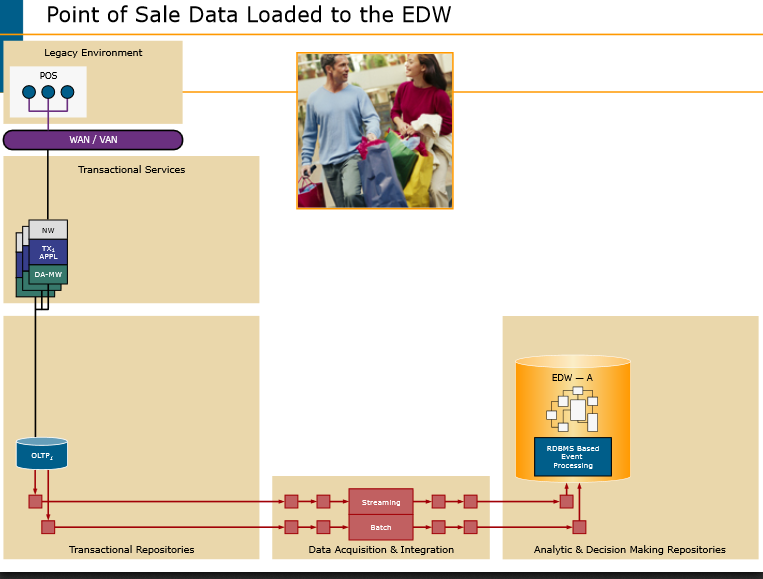
**Data Warehousing**

**Enterprise Data Warehouse**

* **Data mart** is for some of the data for some of the people for some of the time and it can always be finished.
* **Data warehouse** is for all of the data, all of the people for all of the time and it can never be finished.

Application example – DCM – Demand Chain Management

**From POS to Purchase Order** – e.g. Ice Cream Sales Peak in June



*Pos -> WAN - > Transactional Services -> Transational Repo -> Data Acquisition & Intergration –> Analytics & Decision making Repo*

**Scalability – More than just Bytes**

**What is a BIG Enterprise Data Warehouse?**

* **Big table?** 150 billion rows
* **Number of tables?** 300,000
* **Insert/Update per day?** 3 billion records
* **Identified users?** 300,000
* **Queries per day?** 4 million
* **Total spinning disk?** > 40 Petabytes

**Examples of Data Warehouse**

Teradata using

* Hadoop
* Aster
* Teradata bespoke

**Leading industries**

Banking/Financial Services

Government

Insurance & Healthcare

Manufacturing

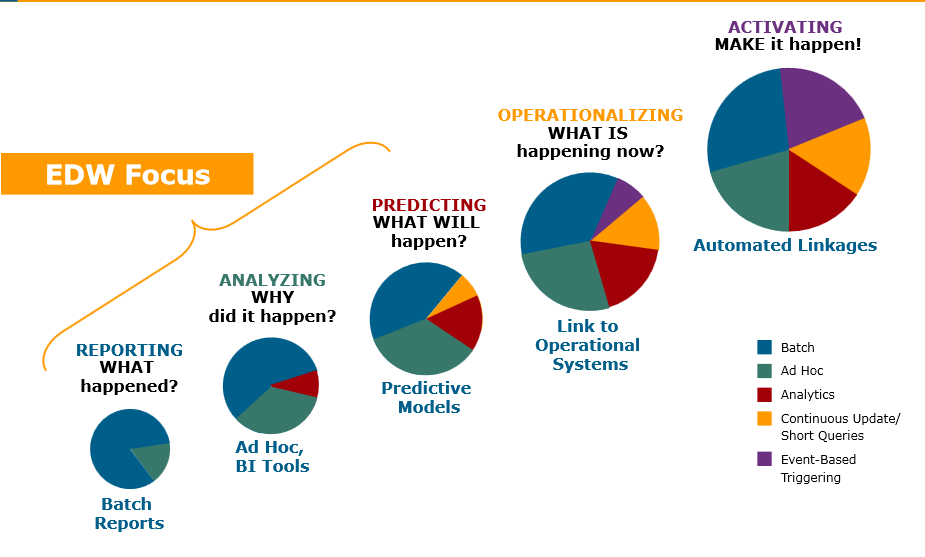
Retail

Telecommunications

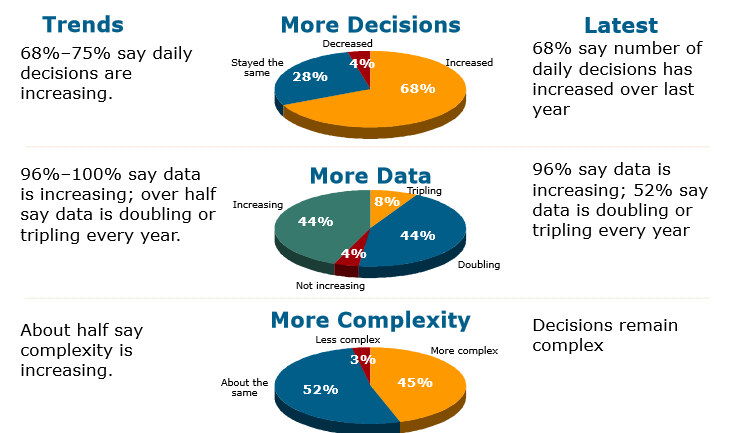
Transportation Logistics

Travel

**EDW Focus**



**Trends Show that**



**Hyper Media**

**What is Hypertext?**

* Hypertext is the organization of information units into connected associations that a user can choose to make.
* An instance of an association is called a link or hypertextlink.

**What is HyperMedia?**

Hypermedia, a term derived from hypertext,

* + extends the notion of the hypertext link to include links among any set of multimedia objects, including sound, motion video, and virtual reality.
  + It can also connote a higher level of user/network interactivity than the interactivity already implicit in hypertext.

**Hypermedia & Hypertext**

Free browsing through information:

* + **Browsing vs Navigation**
    - Browsing is unstructured
    - Navigation has a path structure
  + Browsing is looking at books on shelves to see what is there
  + Navigation is following the structure to locate a book on the correct shelf

**How to get data out of the web**

**Navigation - REST** – Web Directories

* **Manual** ( By Humans ) – Curetted Collection
* **Biased, Time consuming, Changing**

**Browsing - Search Engines**

* piece of software that knows or can work out how relevant or useful a webpage is
* Automatic

**Structure Enables Browsing**

**Think of a newspaper**

* + - **Sections:** Sports, Home News, International News
    - **Stories:** Headlines, Summary Paragraphs, Story Paragraphs
  + **Index**

**Computer Enabled Browsing**

Think of a Hypercard

* + Structered Information Store
  + Free Text Searching based Browsing
  + History of Browsing

**World Wide Web**

**A hybrid Hypermedia System**

* + **Free Browsing**
    - Structure within nodes of information
    - Search Engines
    - History
  + **Navigation**
    - Links

**Problems with Hypermedia**

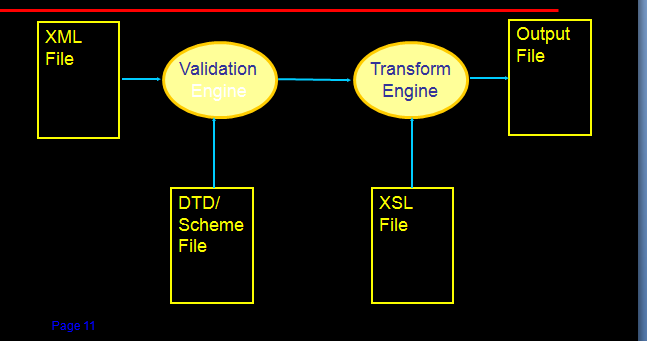
**Content bound with Presentation**

* + **In a database content and presentation are distinct & separate**
  + **So separate content & information**
* **Information is vast & unstructured**
  + **So adapt to the user’s browsing**
  + **So interpret information**

**XML - Extensible Markup Language**

* A system for tagging information within user defined structures
* IS NOT a database
  + No database engine
* IS a method for structuring information
* IS an intermediate form of holding information

**XML System**

****

**Meaing in the Web – 6 Billion + Pages**

* + **Use AI to look through web pages and deduce meaning?**
    - **Very Difficult!!**
  + **Agree meaning of words in meta data**
    - **Need a machine enactable method of searching for material on topics based on meaning.**

**Resource Description Framework**

* A **Resource** is anything that can have a URI; this includes all the Web's pages, as well as individual elements of an XML document. An example of a resource is a draft of the document you are now reading and its URL is http://www.textuality.com/RDF/Why.html
* A **Property** is a Resource that has a name and can be used as a property, for example Author or Title.
* A **Statement** consists of the combination of a Resource, a Property, and a value. These parts are known as the 'subject', 'predicate' and 'object' of a Statement. An example Statement is "The Author of http://www.textuality.com/RDF/Why.html is Tim Bray.”

**<rdf:Description about='http://www.textuality.com/RDF/Why-RDF.html'>**

**<Author>Tim Bray</Author>**

**</rdf:Description>**

* **Hypertext** - Hypertext is basically the same as regular text - it can be stored, read, searched, or edited - with an important exception: hypertext is text with pointers to other text. The browsers let you deal with the pointers in a transparent way -- select the pointer, and you are presented with the text that is pointed to.
* **Hypermedia** - Hypermedia is a superset of hypertext. Hypermedia documents contain links not only to other pieces of text, but also to other forms of media - sounds, images, and movies. Images themselves can be selected to link to sounds or documents. This means that browsers might not display a text file, but might display images or sound or animations. Hypermedia simply combines hypertext and multimedia.

**Some examples of Hypermedia might be:**

* You are reading a text on the Japanese language. You select a Japanese phrase, then hear the phrase as spoken in the native tongue.
* You are viewing a company's floor plan, you select an office by clicking on a room. The employee's name and picture appears with a list of their current projects.

**Benefits**

* Building more maintainable systems
* Decoupling between clients and servers

**Image Based Content Retrieval**

**OLAP VS OLTP**

**MOLAP (multidimensional OLAP)**

tools utilize a **pre-calculated** data set, commonly referred to as a data cube, that contains all the possible answers to a given range of questions.

MOLAP tools are best used for users who have **"bounded" problem sets** (*they want to ask the same range questions every day/week/month on an updated cube, e.g. finance).*

**Advantages**

* **very fast response**
* ability to **quickly write back data into the data set** (budgeting and forecasting are common applications).
* **Can perform complex calculations**: All calculations have been pre-generated when the cube is created

**Disadvantages**

* **Limited scalability** (the cubes get very big, very fast when you start to add dimensions and more detailed data),
* **Inability to contain detailed data** (you are forced to use summary data unless your data set is very small)
* **Load time of the cubes**.
* **Requires additional investment:** additional investments in human & capital resources needed.

**ROLAP (relational OLAP)**

tools **do not use pre-calculated data** cubes. Instead, they intercept the query and pose the question to the standard relational database and its tables in order to bring back the data required to answer the question.

ROLAP tools are best used for users who have "**unbounded" problem set** (they don't have any idea what they want to ask from day to day; e.g., marketing).

**Advantages of ROLAP tools are**

* **ability to ask any question** (you are not limited to the contents of a cube)
* Ability to **drill down to the lowest level of detail** in the database.

**Primary downsides of ROLAP tools are**

* **slow response**
* **some limitations on scalability** (depending on the technology architecture that is utilized).

**HOLAP (hybrid OLAP)**

addresses the shortcomings of both of these technologies by combining the capabilities of both approaches. HOLAP tools can utilize both **pre-calculated cubes and relational data sources**

**OLTP**

**Primary Focus:** *Capturing* and *storing* particular interaction between customer and organisation.

* Interested in a **small number of records** - *Driven by very few.*
* Primarily used to **Insert, Updating, Deleting**
* Records Transactions – Records a fact of importance

**ERP** – Enterprise

* **CRM** – Customer Resource Management

Customer who placed some orders and receives information about order.

Customer TBL – Order TBL – Product TBL

New customer – Add new Customer

New Order – Add new order to order table

**OLAP – Online analytical Processing**

Extracting and Using Data for analytical purposes

* Reports, Dashboard
* Extract Data and turn it into Information
* Make Business Decision from data stored

Customer. Product, Year, Orders,

Summarises and aggregate it under one table – Yearly.

**What is OLAP?**

* Specialised Data Storage ( MOLAP)
* Dimensional analysis Techniques
* A critical component of most business intelligence systems.
* Allows managers to see what is happening to make the best decisions

**Dimensions**

* **Dimensions** - Used to organise data
* **Dimension Members** – Described, Categories into Hierarchies Products, Timeframes

Represented by 3 arrors to represent 3D

**Data Cubes**

* Multidimensional arrays of data
* Data arrays are index by functional categories
* Data cubes are like Data marts with a cell created for every combination of dimension members

**Sparce Datacube** – Some cells can be empty

Benefits of OLAP

Show how business results vary as a function of key drivers

Give the analysist the ability to create and format their own report

Allows the analysis to choose level of detail

**Differences**

OLTP has to summarise 3 tables – join, select, each time

OLAP – select/read – Once to sort yearly.

OLTP – Efficient for small writes

OLAP – Efficient for big reads.

OLTP – Tables are normalised – Easy to find a single customer

OLAP – Unnormalised – Customer may be repeated multiple times

Writing is very sensitive to integrity – OLTP provides roll backs

Better to roll back and say the order was never placed than to have half an order.

OLAP – Requires less transactional functionality

* How file system is designed at the OS level
* What is expected out of them
* The design

BI and Big Data pushes volume extraction by dividing

**Data Mining**

**Data Mining**

**Data Mining Algorithms**

**Data Mining – SQL Server**