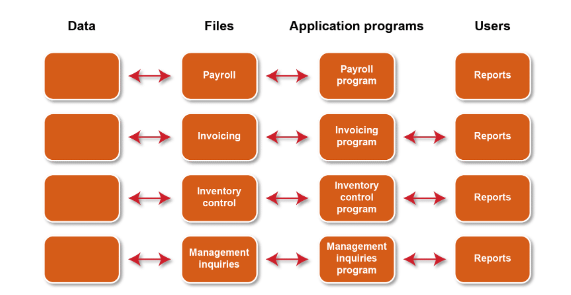
**Unit 9 Lecture cast- Database Models and Systems**

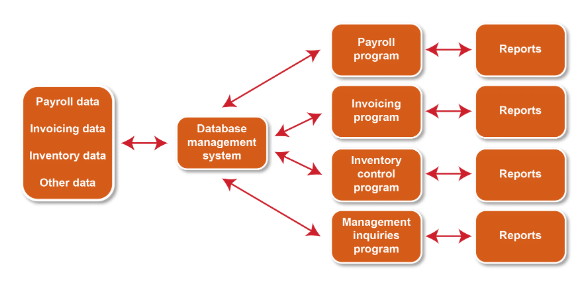
**Approaches to Data Management**

**Traditional approach** is one system for each process, e.g one for HR, one for payroll. Similar to an old filing cabinet system.

Data can be repeated in multiple systems and requires updates in multiple systems, data integrity suffers as a result.

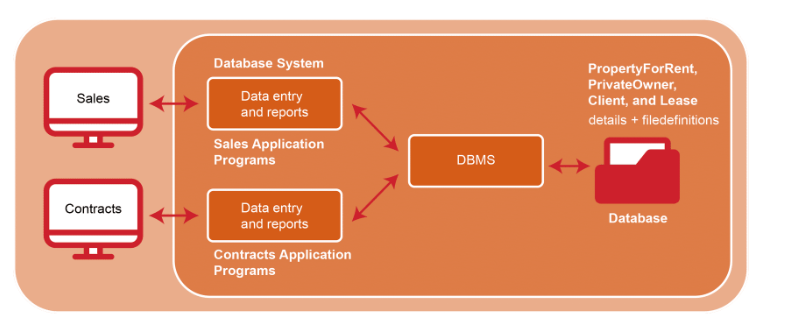


**Database approach –** pool of related data is shared by multiple systems / departments. Makes data sharing and updating easier but requires a database management system to successfully implement.



**Database Management Systems (DBMS)**

**“Enable users to define, create, maintain, and control access to the database”**

****

**A data base application program interacts with the database by issuing an appropriate request (SQL) to the DBMS.**

**DBMS environment has three components:**

* **Hardware – One computer or a network of them.**
* **Software – DBMS, operation, network applications and replated applications for input etc**
* **Data – used by organisation and a description of it (Scheme)**

**DBMS advantages:**

* Control of data redundancy
* Data consistency
* More information from the same amount of data
* Sharing of data
* Improved data integrity
* Improved security
* Enforcement of standards
* The economy of scale
* Balance conflicting requirements
* Improved data accessibility and responsiveness
* Increased productivity
* Improved maintenance through data independence
* Increased concurrency
* Improved backup and recovery services

**DBMS disadvantages:**

* Complexity
* Size
* Cost of DBMS
* Additional hardware costs
* Cost of conversion
* Performance
* Higher impact of a failure

**There are many and varied applications for a DBMS system across all sectors.**

**Database models**

**How a database is implemented.**

**The hierarchical model has fewer complex relationships so easier to manipulate. This makes it efficient when in use but difficult to change and install.**

**The network approach is considered to be flexible, but it is also difficult to develop due to complex relationships between entities.**

**The relational model is the most widely used and most intuitive. Can link relational tables and view data from different perspectives.**

**Database design considerations**

1. **Content – what data will be stored and content**
2. **Access – security and who has access to what data**
3. **Logical structure – assembling data so it makes sense to the user**
4. **Physical organisation – where the data will be stored physically.**

**Data Modelling and Entity Relationship Diagrams (ERD)**

**Physical – where data will be organised and located – for example which server to be used**

**Logical – Model of how database structured to meet requirements. Identifies relationship between data items and expresses these with an entity relationship diagram.**

**Database Implementation**

**Databases are implemented by a DBMS software package.**

**They must support SQL which is a language for extracting data from a database.**

**They must support multiple and concurrent users.**

**Must be able to handle large datasets and be able to migrate between hardware.**

**Examples – MySQL, Microsoft Access, Microsoft SQL Server, Oracle Server.**

**Database Modelling – ERDs**

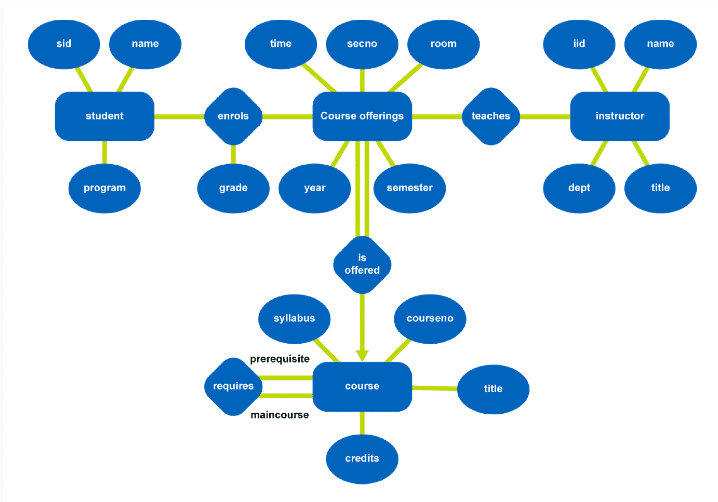
**An ERD diagram shows the entities (tables) and how they join together (relationships). It shows the relationship types and the uniqueness of the values in a column (cardinality) and cardinality of relationships.**

**Relationships can be one to one, one to many, many to many depending on the dependency.**

**Example given of employee in a department, there is a many to one relationship as many employees to a department (this is the cardinality information).**

**Many to many can have issues and require normalisation to remove redundant data and simplify relationships.**

**Example provided for university course database from Stair, 1996):**

****

**Database Implementation Artefacts**

**The following should be produced as a result of implementing a database**

**Schema:**

**Information about database referenced by DBMS. Contains user restriction information to help access and security. It is normally created with a data definition language describing physical and logical paths to the database.**

**Data Dictionary:**

**Detailed descriptions of all data including names, other names to identify it, range of values and type, storage info.**

**Relational model**

**Logical structure designed with a relational model, how data is structured. Different applications can access as a result.**

**Physical structure is mostly hardware.**

**Relation is a table, attribute is a column of a relation (or table).**

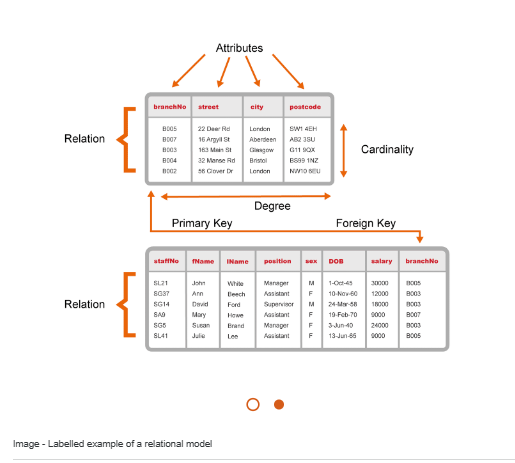
**Domain is allowed values for an attribute (column).**

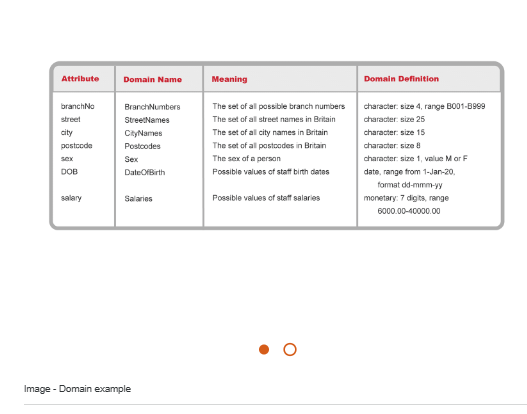
**Tuple is a row of a relation (table row).**

**Degree is how many columns (attributes) in a table (relation).**

**Primary key is a candidate key to identify a unique tuple (row) in a relation (table). Foreign key is an attribute (column) or set of attributes that match a candidate key (primary key in another table).**

**A relational model understanding is essential for data management applications.**

****

****

**Base table – physically stored in the database compared to a view, which a virtual table generated by the DBMS from underlying tables whenever accessed.**

**Database Views:**

**View is a virtual table that may not exist in the database but is produced on request of the DBMS. A query defines that view and can simplify complex operations on tables. Views / queries allow different users to see customised views of same and different data,**

**A database holds a huge range of data for the organisation and different sub sections will be required for different purposes and access levels.**

**Views can be at application level, dynamic, so allowing individuals to create own for multiple uses. The also allow security of systems to be strong and flexible, so users can only see allowed areas of database.**

**Transaction Management**

**Transaction is a unit of work on a database. Application programs are a series of transactions with non-database processing in between action on data for a purpose.**

**Each process that affects a data item must be performed only once, so this could be an update, the system must make this new value available immediately to all users to ensure consistency.**

**A transaction therefore transforms the Database from one state to another, consistency must be maintained. If successful, data committed and database in a new state. If failure, update or database rolled back to return to previous consistent state.**

**Concurrency**

**Managing simultaneous operations without them interfering with each other. Major aim to allow users to be able to access simultaneously, when viewing this is not a huge issue but when more than one updating can be clashes that lead to data problems and inconsistency.**

**Concurrency control prevents this and ensures data integrity.**

**Concurrency Issues to prevent:**

**Lost update problem - an update is overridden by another user.**

**Uncommitted dependency problem – one transaction can see intermediate results of another before committed.**

**Inconsistent Analysis problem – Whilst one transaction is processing, other updates some of the values that were part of the first transaction, resulting in incorrect outcome of first transaction,**