Entity Relationship Diagrams (ERD)

What is an Entity Relationship Diagram
An entity-relationship model is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often it is a relational database, and its requirements are in a top-down fashion. Diagrams created by this process are called entity relationship diagrams.

History of ER Diagrams
Why an Entity Relationship Diagram
Why have Diagrams?
Common Terminology

History of Entity Relationship Diagrams
ERDs were first introduced by Charles Bachman these earlier diagrams described data structures and were known as “Bachman Diagrams”. Bachman did recognise the need to model at a higher level of abstraction and proceeded to take these designs further. However, it was Dr Pin-Shan (Peter Chen), drawing on work of Bachman and his other colleagues that made these diagrams popular. [Refer to “The Entity Relationship Model and Practical Data Modeling by Steve Hickman 2004, Journal of Conceptual Modeling, April 2004” for more information].

Why an Entity Relationship Diagram
An ERD is a logical representation of a database structure shown in an easy to understand pictorial form. The diagram shows three basic building blocks; Entities, Relationships and Attributes between the Entities. (Turner & Weickgenannt 2009) The pictorial form enables any issues to be identified early on such as: many to many relationships, ambiguities, entities and their relationships, deciding on what data needs to be stored, and identifying the degree of a relationship.

“An ERD shows the main things or entities that have a collection of related data elements (fields) that we want to store, and illustrates the relationship between these entities. This is a high level diagram; some people like to refer to it as a view from 30,000 feet.” (Strategic Finance, Jan 2011)

Accountants find ERDs to be an excellent tool to represent the accounting data and entities in the accounting system, because they are a simple way to analyse the complex relationships between entities in the accounting system. (Turner & Weickgenannt 2009)
Why have Diagrams?

Every organisation has a set of systematic steps they undertake to complete their day-to-day activities, as part of the business process. As these business processes occur, the corresponding data will be collected by the internal accounting information system, and outputted in the form of charts, diagrams, reports and documentations. This output allows the accountant to capture record and analyse accounting data as well as understand the procedures and processes of the business process. It is therefore important that users such as systems professionals and accountants understand the documentation and charts that describe the accounting systems. A picture, or chart, of the system is a concise, complete and easy-to-understand way to analyse a process or system.

One of the important tools used in documenting accounting information systems is designing an ERD (Turner & Weickgenannt 2009)

Building Blocks of the Diagram

ERDs identify entities, which become the records in a database, and the attributes of entities, which become the fields of those records. (Turner & Weickgenannt 2009)

Entities

An entity is the concept used for things recognised as being capable of independent existence that are uniquely identified. Entities are often the names of those things such as a computer, an employee, a customer or a supplier. (Wikipedia).

Attributes

Each entity has attributes, or characteristics which describes that entity. A key attribute is the unique distinguishing characteristics of the entity. For instance a supplier name or supplier number would be an attributes that describe the entity supplier.

Relationships

Relationships are defined as being the data shared between the entities. A relationship captures how two or more entities are related to one another, as well as how they are joined together. There are three types of relationships between entities; one to one, one to many and many to many. ERDs do not show single entities they only show entity sets and relationship sets.
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>ERD Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One to one</strong></td>
<td>Each employee has one personal file. Each personal file belongs to only one</td>
<td></td>
</tr>
<tr>
<td></td>
<td>employee.</td>
<td></td>
</tr>
<tr>
<td><strong>One to many</strong></td>
<td>One supervisor has many employees. Employees have only one supervisor.</td>
<td></td>
</tr>
<tr>
<td><strong>Many to many</strong></td>
<td>Each supplier can sell many items, and each item can be purchased from many</td>
<td></td>
</tr>
<tr>
<td></td>
<td>suppliers.</td>
<td></td>
</tr>
</tbody>
</table>

**Cardinality**

Database structures tend to be complex because entities are related to each other, and these relationships can be complicated – i.e: suppliers and inventory items are two sets of entities that are related because the business buys inventory items from suppliers. Any individual inventory item could be purchased from many suppliers, and any single supplier could sell many different items to the business. Another example is the relationship between customers and orders. Each customer can have many orders, but many orders belong to only one customer. The relationships between entities in an ERD are depicted by a concept called cardinality. Cardinality refers to how many instances of an entity relate to each instance of another entity. Cardinality describes three manners in which entities can relate to each other in the following diagram:
Common Terminology

Schema
The schema shows the tables and their relationship includes the field names with indicators of the fields that will be primary and foreign keys along with the relationships between the fields in the tables. The schema and ERD are shown on the same page.

Data Dictionary
The data dictionary states the properties of each field and enables the Database management system to understand how to store, display, receive, and process the data fields. The Data Dictionary prevents input errors by specifying the data type and the range of valid values for each field.

Process Diagrams
Process diagrams detail the process needed to transform the data into information on the reports.

Software Design Tools
Two software design tools used to help create ERDs are ‘SmartDraw’ and ‘DeZign’ for databases however Word, Excel and Access can also be used to draw the diagram.

Tips for drawing the diagram:

- make sure each entity appears only once
- name each entity, relationship and attribute
- look at the relationships between entities
  - are they necessary
  - are all relationships represented
  - eliminate redundant relationships
  - do not connect relationships to each other
- have the diagram reviewed by each area of the company with ownership of the operations, policies and processes
References Used:


