Entity-relationship design for databases

Entity-Relationship Design is a way to plan the management of information in databases to eliminate duplication of information. ER design tools include ER diagrams such as the following:

![ER Diagram](image)

How is a student transcript prepared by a database system? The transcript is a document containing information about a student, information about course sections, and grade information for the student in these course sections.

Someone managing this data must decide what tables to plan and how the tables intersect. For example, to list a student’s grade in a course, data about the student and about the course section is needed.

A table stores instances of a concept, such as Student or Course section. These concepts with instances are called entities in ER design. Entities are represented by ovals in ER diagrams:

![Entities](image)

Often entities have relationships that should be represented in some way. For purposes of student transcripts, registration information, including grades, is required. The corresponding relationship is “Students register for course sections” (see top diagram).

Example problem:

Use Entity-Relationship Design to design a database to represent three entities, each entity represented by one table with a primary key. The entities are: customers, products, transaction detail. Detail is one instance of a product purchased, possibly along with other products. Describe the relationships among these entities. Create a small Excel table for each entity.

Solution:

This is a problem about customers, products, and transaction details. A transaction detail is an instance of a customer purchasing a product, and expresses the relationship, “Customer purchases product”:

![Relationship Diagram](image)

The first step in solving the problem is to identify the tables (entities). These could be called "Customer," "Product", and "Purchase detail".

The next step is to identify attributes of each entity, including a primary key for each entity. A correct answer would be as follows:

Customer: customer-ID, name, address, phone

Product: product-ID, name, price

Purchase-detail: purch-det-ID, customer-ID, product-ID, date, qty

Notice that each table has an attribute that is an identifier for the entity, similar to a student ID or a course ID. The ID is the primary key of the table.

Finally, each of the above could be turned into a table, so that the Customer table would have four columns: customer-ID, name, address, and phone. In a full solution to the problem, you would create three tables, filling them out with a few sample rows.
<table>
<thead>
<tr>
<th>Students</th>
<th>Course sections</th>
<th>Registrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>stu-ID</td>
<td>name</td>
<td>major</td>
</tr>
<tr>
<td>1234</td>
<td>shaw, d</td>
<td>bus</td>
</tr>
<tr>
<td>4312</td>
<td>smith, d</td>
<td>psych</td>
</tr>
<tr>
<td>5678</td>
<td>cox, t</td>
<td>cs</td>
</tr>
</tbody>
</table>