Database Design Concepts

Design a Relational Database
Creating a properly designed database is time-consuming but essential to ensuring accurate information and opportunity for growth. It is important to invest time into planning a database before creating any objects.

Database Objects
Microsoft Access provides four main objects that can be used to create a relational database:

- **Table**: a list of rows and columns, similar to a spreadsheet, that stores data. Tables may store information about grades, students, products, etc.
  - Record: a table row containing a meaningful and consistent way of combining information
  - Field: a table column containing a single item of information.
- **Query**: answers a question about a table’s data, such as “Which students have a GPA of 3.5 or higher?”
- **Forms**: provides an easy way to create, modify, and navigate table records.
- **Reports**: creates an attractive, easy-to-read format to view table records or query results.

Database Design Process
Use the following steps to plan a well-designed database (before ever opening your database software program).

1. Determine the purpose of the database, including a list of potential queries/reports that will be required.
2. Create a list of fields that will be needed in order to create the queries/reports mentioned in step one. Always break fields down into their smallest form (e.g. a name should be displayed as two separate fields – first name and last name).
3. Organize the fields into separate tables, grouped by subject. This is the most crucial and time intensive step in the process.
4. Determine a field within each table that uniquely identifies the record. This field will be the **primary key** (e.g. PSU ID number, SS Number, etc.)
5. Determine the relationships between the separate tables based on common fields (e.g. a PSU ID number is one table can be joined with a PSU ID number in another table).
6. Normalize the tables.

Normalization Rules
Normalization rules ensure proper database design by reducing redundancy and increasing organization. A database must conform to all three normalization rules.

- **First Normal Form**: a database is in first normal form when every field contains a single attribute value.
  - For example, a credit course scheduling database should not have a field for each section.
  - Table 1: An example database table violating the first normal form.

<table>
<thead>
<tr>
<th>CrsCd</th>
<th>Crs#</th>
<th>Sec1</th>
<th>Sec2</th>
<th>Sec3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOC</td>
<td>100</td>
<td>11023</td>
<td>11056</td>
<td>11045</td>
</tr>
</tbody>
</table>

- **Second Normal Form**: a database is in second normal form when it is in first normal form and all non-key fields are dependent on the primary key.
  - For example, an instructor table with a primary key of InstrID should not have a field containing a course description. No functional dependency exists between a course description and an instructor ID number.

- **Third Normal Form**: a database is in third normal form when it is in first and second normal forms and no function dependencies exist between non-key fields.
  - For example, the Dept and DeptChair fields below are dependent upon each other and not the student-related primary key field. The department information should be stored in a department related tables.

<table>
<thead>
<tr>
<th>PSUID</th>
<th>FName</th>
<th>LName</th>
<th>Dept</th>
<th>DeptChr</th>
</tr>
</thead>
<tbody>
<tr>
<td>9-1000-0001</td>
<td>Tom</td>
<td>Smith</td>
<td>SOC</td>
<td>Joe Jones</td>
</tr>
</tbody>
</table>

Table 3: An example database table violating the third normal form.