The Relational Model

This model is simple and it has all the properties and capabilities required to process data with storage efficiency.

Concepts:

1- Tables

The relational model used the basic concept of a relation or table. A database is composed of multiple tables. A table has rows and columns, where rows represent records and columns represent the fields.

2- Columns

A database stores pieces of information or facts in an organized way. The principal storage units are called columns or fields or attributes.

3- Records

Records contain fields that are related. A single row of a table, which contains a single record for that relation is called a tuple.

4- Domain

Domain is a set of acceptable values that a column is allowed to contain. This is based on various properties and the data type for the column.
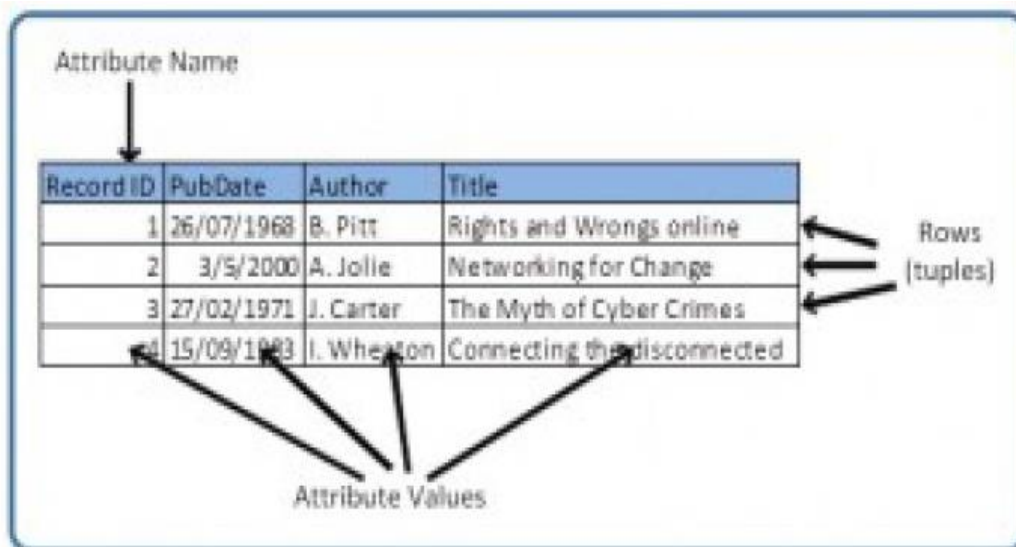

Figure (1): Example of a simple table

Records and fields form the basis of all databases. The simple table example in Figure 1 shows us how fields can hold a range of different sorts of data. This one has:

 • *A Record ID field: this is an ordinal number; its data type is an integer.*

• *A Pub Date field: this is displayed as day/month/year; its data type is date.*

• *An Author field: this is displayed as Initial. Surname; its data type is text.*

• *A Title field text: free text can be entered here.*

 For example, can request that a selection of records be limited by date: 1. all before a given date, 2. all after a given date or 3. all between two given dates. Other example, can choose to have records sorted by date. Because the field, or record, containing the data is set up as a Date field, the database reads the information in the Date field not just as numbers separated by slashes, but rather, as dates that must be ordered according to a calendar system.

5- Relation key

Each row has one or more attributes, known as relation key, which can identify the row in the relation (table) uniquely. Often, keys are used to join data from two or more relations.

Properties of Relational Tables
1-A table has a name that is distinct from all other tables in the database.
2-Values are atomic.
3-Each row is unique.
4-Each column has a unique name.
5-Each column in a relation contains values from a same kind.
6-The sequence of columns is insignificant.
7-The sequence of rows is insignificant.

Integrity Rules and Constraints

Constraints are a very important feature in a relational model. Constraints are useful because they allow a designer to specify the semantics of data in the database. Constraints are the rules that force DBMSs to check that data satisfies the semantics.

Domain Integrity

Domain restricts the values of attributes in the relation and is a constraint of the relational model. However, there are real-world semantics for data that cannot be specified if used only with domain constraints. We need more specific ways to state what data values are or are not allowed and which format is suitable for an attribute. *For example, the Employee ID (EID) must be unique or the employee Birthdate is in the range [Jan 1, 1950, Jan 1, 2000].*

Entity integrity

To ensure entity integrity, it is required that every table have a primary key. Neither the PK nor any part of it can contain null values. This is because null values for the primary key mean we cannot identify some rows. *For example, in the EMPLOYEE table, Phone cannot be a primary key since some people may not have a telephone.*

Referential integrity

Referential integrity requires that a foreign key must have a matching primary key or it must be null. This constraint is specified between two tables (parent and child); it maintains the correspondence between rows in these tables. It means the reference from a row in one table to another table must be valid.

Examples of referential integrity constraint in the Customer/Order database of the Company:

• *Customer(**CustID**, CustName)*

• *Order(**OrderID**, CustID, OrderDate)*

To ensure that there are no orphan records, we need to enforce referential integrity. An orphan record is one whose foreign key FK value is not found in the corresponding entity – the entity where the PK is located. that a typical join is between a PK and FK.

The referential integrity constraint states that the customer ID (CustID) in the Order table must match a valid CustID in the Customer table. Most relational databases have referential integrity. In other words, when the tables are created the referential integrity constraints are set up.

When setting up referential integrity it is important that the PK and FK have the same data types and come from the same domain, otherwise the relational database management system (**RDBMS**) will not allow the join.

RDBMS is a popular database system that is based on the relational model. Relational database systems are easier to use and understand than other database systems.

Referential integrity in Microsoft Access

In Microsoft (MS) Access, referential integrity is set up by joining the PK in the Customer table to the CustID in the Order table. See Figure 2 for a view of how this is done on the Edit Relationships screen in MS Access.