

CREATING AN ERROR-FREE TABLE IN DATA BASE SYSTEM

In this article explains how to create a table, fields to a table, to set a table's primary key, to set field and table properties. Database objects depend on tables; it should always start design of a database by creating all of its tables and then creating any other objects.

Key words: primary key, foreign key, tracks, data base design basics, default view;

Ч.Г. Сатыбалдиева
преподаватель каф."КЛ" ОшТУ

СОЗДАНИЕ ТАБЛИЦЫ БЕЗ ОШИБОК В СИСТЕМЕ БАЗЫ ДАННЫХ

В статье описывается процедура создания таблицы в системе базы данных, создать таблицу, добавить поля в таблицу и как установить первичный ключ таблиц, также рассматривается в статье создание свойства полей и таблиц.

Ключевые слова: первичный ключ, внешний ключ, вид по умолчанию, создание и проектирование база данных;

A table is a database object that you use to store data about a particular subject, such as employees or products. A table consists of records and fields. Before you create tables, carefully consider your requirements and determine all the tables that you need. For an introduction to planning and designing a database, see the article Database design basics. [1]

Each record contains data about one instance of the table subject, such as a particular employee. A record is also commonly called a row or an instance. Each field contains data about one aspect of the table subject, such as first name or e-mail address. A field is also commonly called a column or an attribute.

A record consists of field values, such as Contoso, Ltd. or someone@example.com. A field value is also commonly called a fact.

Table 1

Database design basics

Customers				
	ID	Company	First Name	Last Name
	1	Company A	Anna	Bedecs
	2	Company B	Antonio	Gratacos Solsona
	3	Company C	Thomas	Axen

1. A record
2. A field
3. A field value

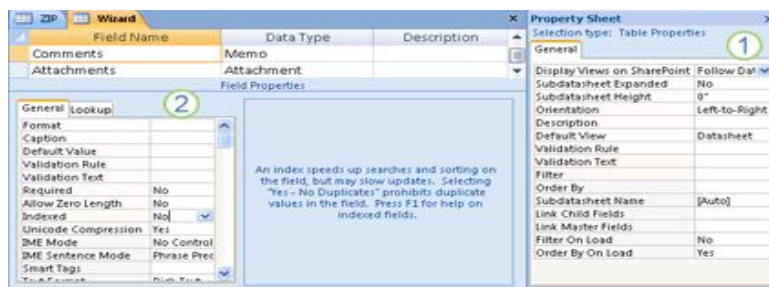
A database can contain many tables, each storing information about a different subject. Each table can contain many fields of different types of data, such as text, numbers, dates, and hyperlinks. [2]

Table and field properties

Tables and fields also have properties that you can set to control their characteristics or behavior.

Table 2

Design view



A table open in Design view.

1. Table properties
2. Field properties

In an Access database, table properties are attributes of a table that affect the appearance or behavior of the table as a whole. Table properties are set in the table's property sheet, in Design view. For example, you can set a table's **Default View** property to specify how the table is displayed by default. [2]

A field property applies to a particular field in a table and defines one of the field's characteristics or an aspect of the field's behavior. You can set some field properties in Datasheet view. You can also set any field property in Design view by using the Field Properties pane.

Data types

Every field has a data type. A field's data type indicates the kind of data that the field stores, such as large amounts of text or attached files.

Table 3

A field properties

Field Name	Data Type	Description
Comments	Memo	
Attachments	Attachment	
Field Properties		

A data type is a field property, but it differs from other field properties as follows:

- You set a field's data type in the table design grid, not in the **Field Properties** pane.[3]
- A field's data type determines what other properties the field has.
- You must set a field's data type when you create the field.

You can create a new field in Access by entering data in a new column in Datasheet view. When you create a field by entering data in Datasheet view, Access automatically assigns a data type for the field, based on the value that you enter. If no other data type is implied by your input, Access sets the data type to Text. If needed, you can change the data type by using the Ribbon, part of the new Microsoft Office Fluent user interface.

Examples of automatic data type detection

The following table shows how automatic data type detection works in Datasheet view.

Table relationships

Although each table stores data about a different subject, tables in a database usually store data about subjects that are related to each other. For example, a database might contain:

- A customers table that lists your company's customers and their addresses.
- A products table that lists the products that you sell, including prices and pictures for each item.
- An orders table that tracks customer orders.

Because you store data about different subjects in separate tables, you need some way to tie the data together so that you can easily combine related data from those separate tables. To connect the data stored in different tables, you create relationships. A relationship is a logical connection between two tables that specifies fields that the tables have in common.

Keys

Fields that are part of a table relationship are called keys. A key usually consists of one field, but may consist of more than one field. There are two kinds of keys:

- **Primary key** A table can have only one primary key. A primary key consists of one or more fields that uniquely identify each record that you store in the table. Often, there is a unique identification number, such as an ID number, a serial number, or a code, that serves as a primary key. For example, you might have a Customers table where each customer has a unique customer ID number. The customer ID field is the primary key of the Customers table. When a primary key contains more than one field, it is usually composed of pre-existing fields that, taken together, provide unique values. For example, you might use a combination of last name, first name, and birth date as the primary key for a table about people.

- **Foreign key** A table can also have one or more foreign keys. A foreign key contains values that correspond to values in the primary key of another table. For example, you might have an Orders table in which each order has a customer ID number that corresponds to a record in a Customers table. The customer ID field is a foreign key of the Orders table.

The correspondence of values between key fields forms the basis of a table relationship. You use a table relationship to combine data from related tables. For example, suppose that you have a Customers table and an Orders table. In your Customers table, each record is identified by the primary key field, ID.

To associate each order with a customer, you add a foreign key field to the Orders table that corresponds to the ID field of the Customers table, and then create a relationship between the two keys. When you add a record to the Orders table, you use a value for customer ID that comes from the Customers table. Whenever you want to view any information about an order's customer, you use the relationship to identify which data from the Customers table corresponds to which records in the Orders table:

1. A primary key, identified by the key icon next to the field name.
2. A foreign key — note the absence of the key icon.

Concluding each item of data is recorded only once, in one table, there is less opportunity for ambiguity or inconsistency. For example, you store a customer's name only once, in a table about customers, rather than storing it repeatedly (and potentially inconsistently) in a table that contains order data. The design of a database is easier to understand if the subjects are properly separated into tables.

Literature:

1. **Bieniek, D.**, Technologies for Data Analysis: Data Mining, Visual Mining, Text Mining, OLAP BHV [Текст] / R. Dyess, M. Hotek, J. Loria, A. Machanic, A., Wiernik // Peterburg. A. (2006).
2. **Eck, D.J.** (2006). Introduction to Programming and Creating Table. <http://math.hws.edu/javanotes> [Текст] / Eckel, B. (2001).