**KNRDER**

Data | Encryption | Resources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2354 Main St. Boise, ID. 83725, USA |  | p. 1-208-555-1234 |  | Base@knrder.com  www.knrder.com/base |

KNRDERBase

# Introduction

The goal of KNRDER has always been to deliver a great experience to users in a simple, user-friendly manner. That rationale drives us, Konnor, Nate, and Ryan, in every choice and decision and change we implement into our database systems. We as a team are committed to the needs of every employer. This customer-based mode of thinking has led to the creation that we have dubbed KNRDERBase. This database excels as being a simple way to store all the data any business would need to have in one document. No longer are the days that information is stored across multiple documents in different formats. On top of that, this database is user-friendly with implementation of forms. Instead of going through every database to add new data, we have added forms for a simple and quick way to a large amount of data. Here at KNRDER, we are committed to the customer and in every way our database reflects that philosophy.

Table of Contents

1. [Executive Summary 1](#_Toc409597840)

[Why KNRDERBase?](#_Toc409597841)

[What We Track](#_Toc409597842)

1. [Entity Relationship Diagram 2](#_Toc409597843)

[Description](#_Toc409597844)

[Standard Relationships](#_Toc409597845)

[Notable Relationships](#_Toc409597846)

1. [First, Second, and Third, Normalization Forms 3](#_Toc409597847)

[First Normal Form (1NF)](#_Toc409597848)

[Second Normal Form (2NF)](#_Toc409597849)

[Third Normal Form (3NF)](#_Toc409597850)

1. [Access Implementation 4](#_Toc409597851)

[Sample Tables & Data](#_Toc409597852)

[Sample Forms](#_Toc409597853)

[Sample Reports](#_Toc409597854)

1. [SQL Demonstration 6](#_Toc409597855)
2. [Table of Figures 9](#_Toc409597856)

# Executive Summary

We at KNRDER strive to provide the best possible service to our customers, through KNRDERBase we will continue this tradition of service by providing a simple and functional way to track all of the information important to the effective management of any business. We will provide this service in a way that allows our customers to use it effectively with a minimum of outside help.

## Why KNRDERBase?

We promote efficient and effective data storage though the use of our databases, managed by our effective team of trained database professionals. Through the use of forms we provide easy ways for our customers to view their data, and further to edit existing data and even add new data in a simple, yet effective way. We form our databases with the customers in mind in order to ensure that they get the most effective database for them. To achieve this we strive to provide table relationships that are self-explanatory and simple to use.

## What We Track

In our standard database we track what you need to operate your business in the most efficient way possible. To start out we track employee data to promote easy payroll and tax processing. Second, we track customer and order data to keep track of repeat customers for faster order processing, and easily accessible profit reports with up to date information on incoming and outgoing orders. Third, we track items, inventory, and locations to have accurate and current information on what items are stored where and when it is necessary to restock. Finally we track shippers and suppliers so we can know who and where our goods are being transported to/by. Each of these will be further explained later in this document.

### Employee Data

We track many aspects of each of your employees. This includes information like their contact information, Full name and address, as well as phone number. We also track their payroll information such as their job title, salary, and employment type.

### Customers & Orders

For each person who orders one of our products, we track their shipping information, including their full name, shipping address, and a confirmation email to which to send the digital receipt. We also track much of the order information, such as what they ordered, how much they ordered, and when the order was placed.

### Items, Inventory, & Locations

For every product and service offered by our company we track its name, sale price, cost to produce, and who supplies it. We also keep track of how much of each item we have stored at any specified location. Finally we track the full address of our warehouses, their contact information and manager ID, and the shipping company that works with the location.

### Shippers & Suppliers

We track the full contact information of each of our shippers and suppliers, as well as the locations of the world that they service.

# Entity Relationship Diagram

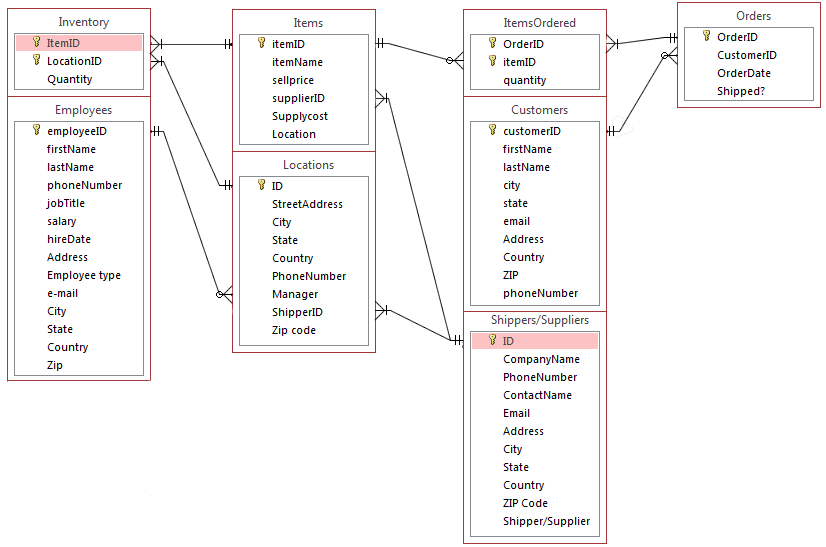


Table 1: Example Entity Relationship Diagram for Idaho Outdoor Adventures

## Description

An Entity Relationship Diagram or ERD is a graphical representation of how a database fits together and how each table interacts with the others. In the diagram shown above you can see the ERD for the first implementation of KNRDERBase, through our partners, Idaho Outdoor Adventures. Described below are the standard and notable relationships shown by the ERD above.

## Standard Relationships

The majority or the relationships in this ERD are the usual One to Many relationships and take place between a primary key and a foreign key of the same name, for example, CustomerID in the Customers table, and the CustomerID in the Orders table. Most small databases will only need this kind of relation to be effective, we at KNRDER however take the effectiveness a step further with a few more notable relationships.

## Notable Relationships

Some of the relationships shown in the diagram above are noticeably different from the standard One to Many relationships that many database users are used to. One example is our relationship between the employeeID field of the Employees table, and the Manager Field of the Locations table. This is a One to Zero Or Many relationship and has the added notability in Access that, when the tables are combined, will show all of the locations and only the employees listed as managers.

# First, Second, and Third, Normalization Forms

Normalization of the customer’s database is a state in which benefits increase with efficiency—a service that we provide to the customer. For an example of how KNRDERBase implements the requirements for all three forms of Normalization, turn to the Access Implementation section of this document.

## First Normal Form (1NF)

When a table is in 1NF, all the data values are atomic, in that the data cannot be split apart into separate fields. For example, a 1NF table would have a first name and last name field, instead of just a ‘name’ field. The next qualification for 1NF is that there is to be no repeating fields, or having the same data value for many fields in a table, so that SQL programming is less cumbersome to the engineers, and confusion far less apparent. Next, each field must have a unique name, so as to be easily distinguished from the other fields. Finally, there must be an ID for each entry, so as to differentiate identical entries.

## Second Normal Form (2NF)

2NF is an expanded normalization of 1NF which only applies to tables with a composite or dual key. A table must first meet the qualifications for 1NF and meet the additional requirement in the case of composite keys that every attribute of the table is directly related to every part of the composite key. If the table only has one key field, it is automatically in 2NF.

## Third Normal Form (3NF)

3NF is an expanded normalization of 2NF. A table in 2NF (and only 2NF) is qualified for 3NF status if all non-key attributes are only dependent on the key fields and no others, with the exception of a calculated field, those however should only show up on reports, and not in the tables.

Consider the following section from our Shipper/Supplier table:

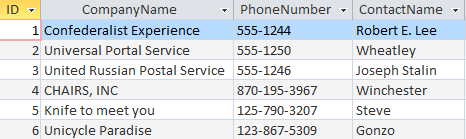


Table 2: Example of a 3NF table

As it can be seen, there are no repeating values, no multiple unique fields, and all fields are atomic, the primary key being the ID field. This qualifies for 1NF

Furthermore, all attributes are dependent on the primary key, and all non-key attributes are only dependent on the keys, and not on any other attributes; this qualifies for 3NF.

# Access Implementation

While we may explain the database and tell you, the customer, of all its greatness, the concern rises over practical application of the database. This is what the next section entails. We have selected a few tables, forms, and reports to show you that we are true to our word on keeping the database simple and efficient.

## Sample Tables & Data

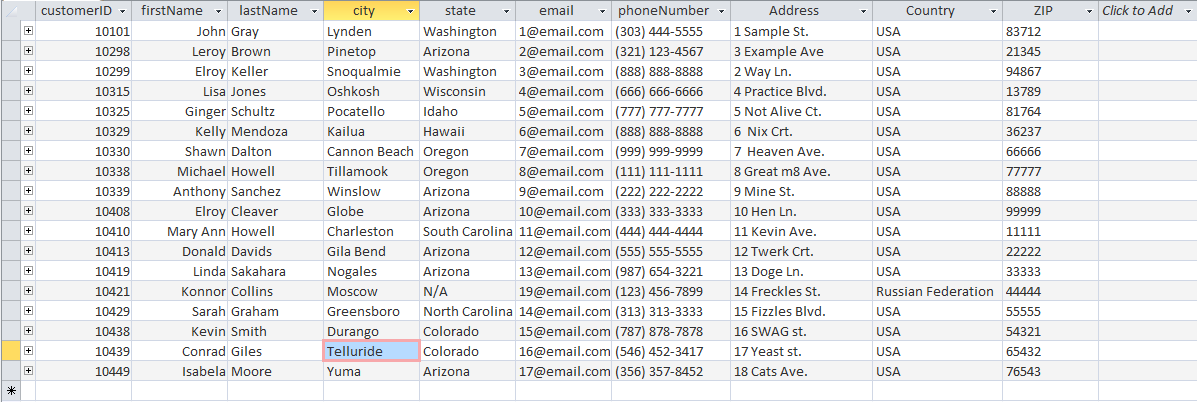


Table 3: The Customers table for Idaho Outdoor Adventures

The sample we have selected is our customers table. This customer table provides all of the data you would ever need on your customers; their name, their exact address, and two ways to contact them if need be. If there is more data needed, there’s a button shown on the upper right corner called “Click to Add” which adds a new field where any further data you would need is provided. For now, we have provided the basics that every company needs to succeed.

## Sample Forms

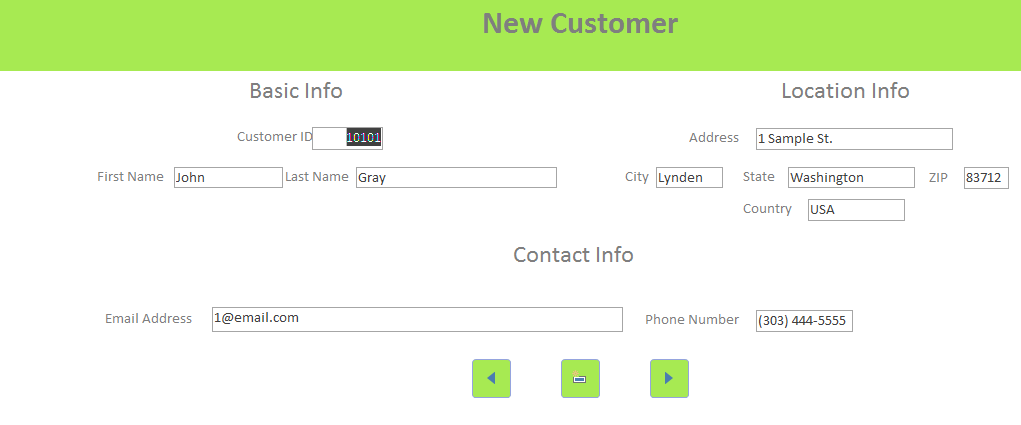


Table 4: The new customer entry form for Idaho Outdoor Adventures

Forms provide an easy way to manage data. The customer table is not the most user-friendly way to create new data, and our form solves that. We have split the data into 3 groupings: Basic Info, Location Info, and Contact Info. Our Basic Info shows just their first name and their unique customer ID that will be used inside the database. The Location Info is set up like an address on a letter to make entries much friendlier and less intimidating than a row of blank fields. Our contact info provides the ways in which a customer could be contacted: Their e-mail and phone number.

## Sample Reports

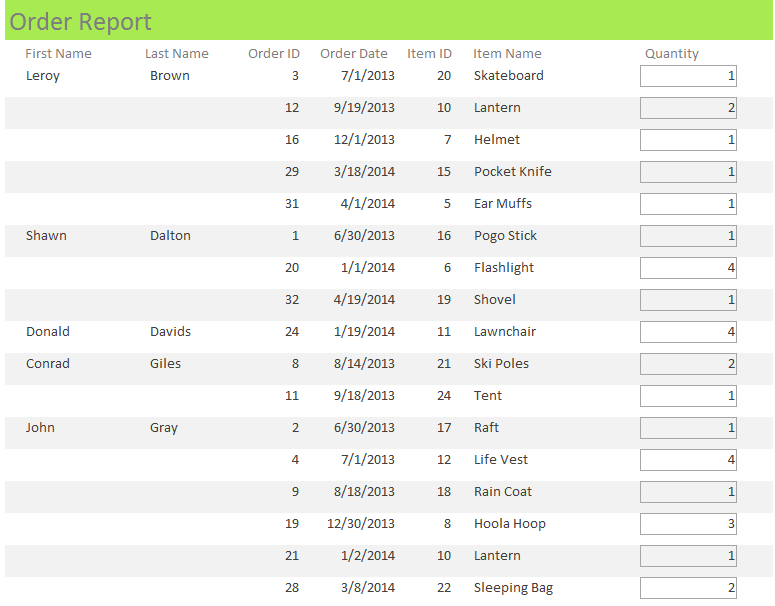


Table 5: The order report for Idaho Outdoor Adventures

This report is the order report, which shows who ordered what and how much they ordered and when it was ordered. This shows the power of KNERDERBase at its fullest. We have made ways to manipulate data outside of tables and make reports on just about anything. Everything is related in our database, and any report can suit the needs of any business.

# SQL Demonstration

Queries are a powerful tool for any database designer. Queries, using other products besides KNDERBase, is difficult because a person must have extensive knowledge in SQL. With our database, our pre-made queries cover just about everything a business could need. Any additional queries can be made using Access’ simple query design methods.

On top of that, queries can be used to generate reports. Reports provide an easier way to see the same data in an easier to understand matter.

Our first demonstration of the power of queries is our inventory location query. The SQL for it is as follows:

SELECT StreetAddress, City, State, [Zip Code] AS zc, Country

FROM Locations

WHERE Country =( SELECT Country FROM Locations WHERE City LIKE 'Boise');

In this example, we are trying to find all of our warehouse locations that exist within the United States. Since Boise is located within the United States, we using SQL to find all cities that have the same country as Boise (USA). The query returns these warehouse locations:

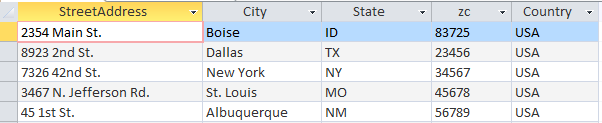


Table 6: Warehouse locations in the USA

Next, we want to look at each location’s inventory. We have this set up as a parameter query, so we can filter the data and not have a specific query for each warehouse location. For this example, we use the Boise location. The main problem here is that the data we need comes from two different locations: we need the item name, which exists only in the items table, and the quantity from the inventory table. To do this, we needed to use a SQL join. A SQL join states how two tables are related and the field by which they are. In the Boise example, we need to link two tables: inventory and items, while filtering the results to only display the Boise location. The SQL for this is as follows:

SELECT i.itemName, inv.quantity

FROM Items AS i INNER JOIN Inventory AS inv ON i.itemID=inv.ItemID

WHERE inv.LocationID=(SELECT ID FROM Locations where ID=[Enter Location ID]);

What the SQL query is doing is joining the two tables, forming a relationship between the Primary key “itemID” in Items table and the foreign key “Item.ID” from the inventory table. After that is accomplished, we needed to filter the results to only show the Boise location. To do this, we ran a sub query that selects quantity based on a parameter entered by the user. For this, we entered the Boise warehouse ID in order to filter the data to only show the Boise inventory.

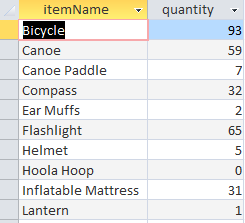


Table 7: The Boise warehouse join query

# Table of Figures

[Table 1: Example Entity Relationship Diagram for Idaho Outdoor Adventures 2](#_Toc408994416)

[Table 2: Example of a 3NF table 4](#_Toc408994417)

[Table 3: The Customers table for Idaho Outdoor Adventures 4](#_Toc408994418)

[Table 4: The new customer entry form for Idaho Outdoor Adventures 5](#_Toc408994419)

[Table 5: The order report for Idaho Outdoor Adventures 6](#_Toc408994420)

[Table 6: Warehouse locations in the USA 7](#_Toc408994421)

[Table 7: The Boise warehouse join query 7](#_Toc408994422)