# **COMP-1701 - Transferring Data to Databases**

## **Data Science & Machine Learning (DSML) - RRC Polytech**

Module E.2 - Entity Relationship Diagram (ERD) – Add More “people” Metadata Columns

When building tables, discussions must be had, for various business needs and logic that follow the digital workflow of the company. We employ naming conventions to keep table objects together alphabetically (as you will see soon in your table listings).

First we need to discuss the people table, and what other fields we need.

**people - all**

We are building this database to be a website, so all websites would have a login page, using an email address and password.

For email address, roughly about 50 characters with variable widths should be used, so a VARCHAR(50) would suffice.

For password, we will use the build in MD5 function to create the password. Each password will be a 32 character constant length so a CHAR(32) would be required.

The adjustment to our ERD, should be as follows (remember to add a column, hover over the column spot where you want the new field to be below, and hover to the right where the blue arrow is to add the new field):  
  
A screenshot of a computer program

Description automatically generated

**people - all**

What other fields? Thinking contacting and locational, so what could those fields be?

**people – all – phone number**

We have the email address, though most of us have **phone number(s)**, so we need to add that, whether it’s an employee, customer, or external company representative. Likely saving a few fields for this. The primary phone number any person, should be mandatory, thus NOT NULL. Though we would need to apply this later, when adding that data, we have existing people rows and so the NOT NULL needs to be applied to the table later.

Though we can mark it as such now in the ERD:

A screenshot of a computer

Description automatically generated

**people – all – address/delivery information**

All of the people we are tracking will have address information, including:

* building types (addr\_type)
  + building, apartment, townhouse, warehouse, etc.
* address prefix (addr\_prefix, for suite, floor, apartment, building number)
* street address (addr)
* postal/zip codes (addr\_code)
* town/city name (town\_city\_name)
* province/state/prefecture… (prov\_state\_pref)
* and country name (country\_name).

Some people will have PO Box information.

Customers will also require some delivery information. Warehouses for the items we sell, may have some delivery related information too.

A graph paper with text and arrows

Description automatically generated

**people – what else? Administrative/System Fields!**

We would need some quick fields, tracking mechanisms, that the system could have, such as a BIT field for an **employee** record, 1 if they are, 0 if they are not.

When entering the information to the system, the employee’s p\_id will be placed in the session memory of the website, after logging in with their email and password, so a **userid\_add**, **date\_add**, and **userid\_mod**, **date\_mod** field, that indicates which **p\_id** and **on what date**, the record was first **add**ed and whom had subsequently **mod**ified the people record in the system.

Lastly, we add an **active BIT** field, which gives us the flexibility to disable the person by switching active=0, then their account is deactivated and by default when adding the person, their active will be 1, meaning they are enabled in the system. In other tables, the active BIT is used to show or hide data. So using WHERE active=1 on the public site/system, is a must to be able to login OR see an item.

A grid with text on it

Description automatically generated

**people – are we done?**

No, this is an awfully big table and we need to ask ourselves:

* Can we maintain UNIQUE records now?
  + Which columns can we use as our UNIQUE key? (4)
  + What other columns alone, need to be UNIQUE, aside from the p\_id? (1)
* Which other columns will contain the duplicates? (4-7)
  + How can we address that? (4 new tables)
* Are any columns within going to be NULL for most of their entries? (?)

**Next up…**

Get this version of your table complete, and next class, we will discuss UNIQUEness and acceptable/unacceptable DUP(LICAT)ES!

## 