According to the backend code in “C:\Projects\_Dev\Websites\AdminExtendedNauEdu\SiteRoot\Prospects\ProspectManagement\ProspectProspectCleanup.aspx.cs”, the strong match which would be an exact match is 0.0.

public const double EXACT\_MATCH = 0.0;

public const double STRONG\_MATCH = 0.20;

public const double MEDIUM\_MATCH = 0.40;

public const double WEAK\_MATCH = 0.65;

public const double NO\_MATCH = 1.0;

public const double NO\_INFO = -1.0;

This conclusion is strengthened in the SQL statement in the same class file that states “DECLARE @match\_score AS FLOAT

SET @match\_score = 0 --Default match\_score is a 'perfect' match.”.

SELECT \* FROM [Distance].[dbo].[Person\_Match\_Comparison] gave me 2,395,563 records. This includes each and every single record in the table.

**How do I find the count of the matches with the highest weight (most in common)?**

SELECT \* FROM [Distance].[dbo].[Person\_Match\_Comparison] WHERE match\_score = 0; gave me 424,168 records. These are records that theoretically are exact matches. This may include multiple records of the same comparison of two people though.

**Not including matches that have already been evaluated**

I looked in the same class file, ProspectProspectCleanup.aspx.cs. On the 180th line, the “Not Duplicate” button method is declared. It updates rows and sets multiple variables shown below.

UPDATE [Distance].[dbo].Person\_Match\_Comparison

SET human\_match\_date = GETDATE(),

human\_match = @isMatch,

human\_match\_uid = @uid

WHERE id = @id "

Below the SQL statement, “@isMatch” is default false. This means whenever someone clicks the Not Duplicate button, their “human\_match” column is set to false. I believe from this logic, we can safely say that the “human\_match” column is null if the comparison hasn’t been evaluated yet, 0 (for false) if the comparison has been evaluated as not a duplicate, and 1 (for true) if the comparison has been evaluated as a definite duplicate.

SELECT \* FROM [Distance].[dbo].[Person\_Match\_Comparison] WHERE match\_score = 0 AND human\_match IS NULL; gave me 419,693 records. The previous statement checks comparisons that are theoretically exact matches AND haven’t been evaluated yet. This may still include repeating comparisons.

**Making sure to exclude repeated comparisons**

SELECT DISTINCT record1\_id, record2\_id FROM [Distance].[dbo].[Person\_Match\_Comparison] WHERE match\_score = 0 AND human\_match IS NULL; gave me 242,744 records

This particular SQL statement selects only the two people’s ids being compared, and with the use of DISTINCT, it makes sure the id’s aren’t repeated. This does not include if the id’s switched columns. (Like if one record compared Adam and Jack, then another record later on compared Jack and Adam, I’m not sure how to do that). This statement also keeps the previous search restraints, such as exact match and not evaluated yet.

**What id numbers are in this table? PID? EMPL? Other?**

The “id” columns in the [Distance].[dbo].[Person\_Match\_Comparison] table as well as their definitions are as follows:

Id: This is the comparison id which is unique to every record in the table.

Record1\_id: This is the first person being compared in each comparison. The specific id is the persons Prospect ID.

Record2\_id: This is the second persons Prospect ID. So far, we have the id of the comparison, and the two Prospect ids of the people being compared.

Human\_match\_uid: I believe this is the id of the employee who evaluates the comparison. It looks like it defaults to null until someone selects duplicate or not duplicate.