PK PK M M PD TD PD

Graphical user interface, application

Description automatically generated

* Identify all fields that are multi-valued
  + SRC department
* Identify fields that are transitively dependent, and model what the end-result would look like in a database model
  + TD = src-loc\_code, you would need a bridge table and separate columns departments
* Identify composite fields
  + Location
* How many tables will be in the final model?
  + List those tables (no fields or keys, just tables)
  + Stores, department and locations, department look up
* What ‘Normal Form’ is this data in right now?
  + First normal form because we made child tables
* Sketch out a “Vertabelo” /diagram model of the db (with PKs/FKs, cardinality, data types etc…)
* Explain the logical test you would apply to the data to find transitive dependencies.
  + I took each none key field and asked them against themselves
* How are Multiples ‘resolved’ in a normalization problem (e.g. what do you do to them to get them in good db structure?)
  + Keep stores in parent table
  + Make child table for departments
* How are composite PKs (two PK fields make a row unique) ‘resolved’ in a normalization problem (e.g. what do you do to them to get them in good db structure?)
  + Make two tables, one for stores and one for locations
  + They each have their own primary key and they can relate to each other
* How are transitive dependencies resolved?
  + Through a look-up table
  + You must break the tables down

**Migration Portion:**

**Cleansing questions:**

* I want to update my departments and some of them are out of date: Both ‘Computers’ and ‘Phones’ should be updated to ‘Devices’

UPDATE Store\_Source

SET src\_department\_2 = 'Clothing'

WHERE src\_department\_2 = 'Mens Clothing' Or src\_department\_2='Womens Clothing'

UPDATE Store\_Source

SET src\_department\_1 = 'Clothing'

WHERE src\_department\_1 = 'Womens Clothing' Or src\_department\_1='Mens Clothing'

-- or in targeted format ...

Update Store\_Source

set src\_department\_1='Clothing'

* + where src\_id =1 Or src\_id=7
* ‘Pots and Pans’ should become Housewares

update Store\_Source

set src\_department\_1='Housewares'

where src\_department\_1='Pots and Pans'

update Store\_source

set src\_department\_2='Housewares'

* + where src\_department\_2='Pots and Pans'
* ‘Instead of ‘mens and womens’ clothing, let’s just have ONE ‘Clothing’ Category.

**Transformation Questions:**

1. Write the Code that inserts the Location Codes:

INSERT INTO location\_codes

SELECT DISTINCT src\_loc\_code, src\_loc\_name

FROM Store\_Source

Table

Description automatically generated

1. **Write the Code that inserts the stores:**

INSERT INTO stores

SELECT DISTINCT src\_id, src\_store\_name,

SUBSTRING(src\_hq\_location,1,CHARINDEX(',',src\_hq\_location)-1),

SUBSTRING(src\_hq\_location,CHARINDEX(',',src\_hq\_location)+2,2),

src\_loc\_code

FROM Store\_Source  
Table

Description automatically generated

*Here is a handy function that splits out the City and State:*

SUBSTRING(src\_hq\_location,1,CHARINDEX(',',src\_hq\_location)-1) as 'store\_hq\_city',

SUBSTRING(src\_hq\_location,CHARINDEX(',',src\_hq\_location)+2,2) as 'store\_hq\_state',

1. **Write the code that inserts the departments:**
2. INSERT INTO store\_departments
3. SELECT DISTINCT src\_id, src\_department\_1
4. FROM Store\_Source
5. WHERE src\_department\_1 is not null
6. UNION
7. SELECT DISTINCT src\_id, src\_department\_2
8. FROM Store\_Source
9. WHERE src\_department\_2 is not null
10. GO

Graphical user interface, application

Description automatically generated with medium confidence

1. **Write Verification Code that looks like this.**

Table

Description automatically generated

/\*

SELECT store\_id, store\_name, store\_hq\_city + ', ' + store\_hq\_st\_abbr as 'HQ Location', sd\_dept\_name, loc\_code\_name

FROM stores

JOIN store\_departments on (store\_id = sd\_store\_id)

JOIN location\_codes on (store\_loc\_code = loc\_code)

SELECT sd\_dept\_name, COUNT(sd\_store\_id) as 'Number of Departments of this Type'

FROM store\_departments

GROUP BY sd\_dept\_name

order by COUNT(sd\_store\_id) desc

GO

\*/

select \* from stores

select \* from store\_departments

select \* from location\_codes