Modeling Comparison Between Relational and Document Databases

Relational to Document Database Terminology Translation

RDBMS	MongoDB
Database	Database
Table Entity in Design	Collection
Row Tuple in Design	Document
Column Attribute in Design	Field
Index	Index
JOIN	Embedded document, document references, or `$lookup` to combine data from different collections

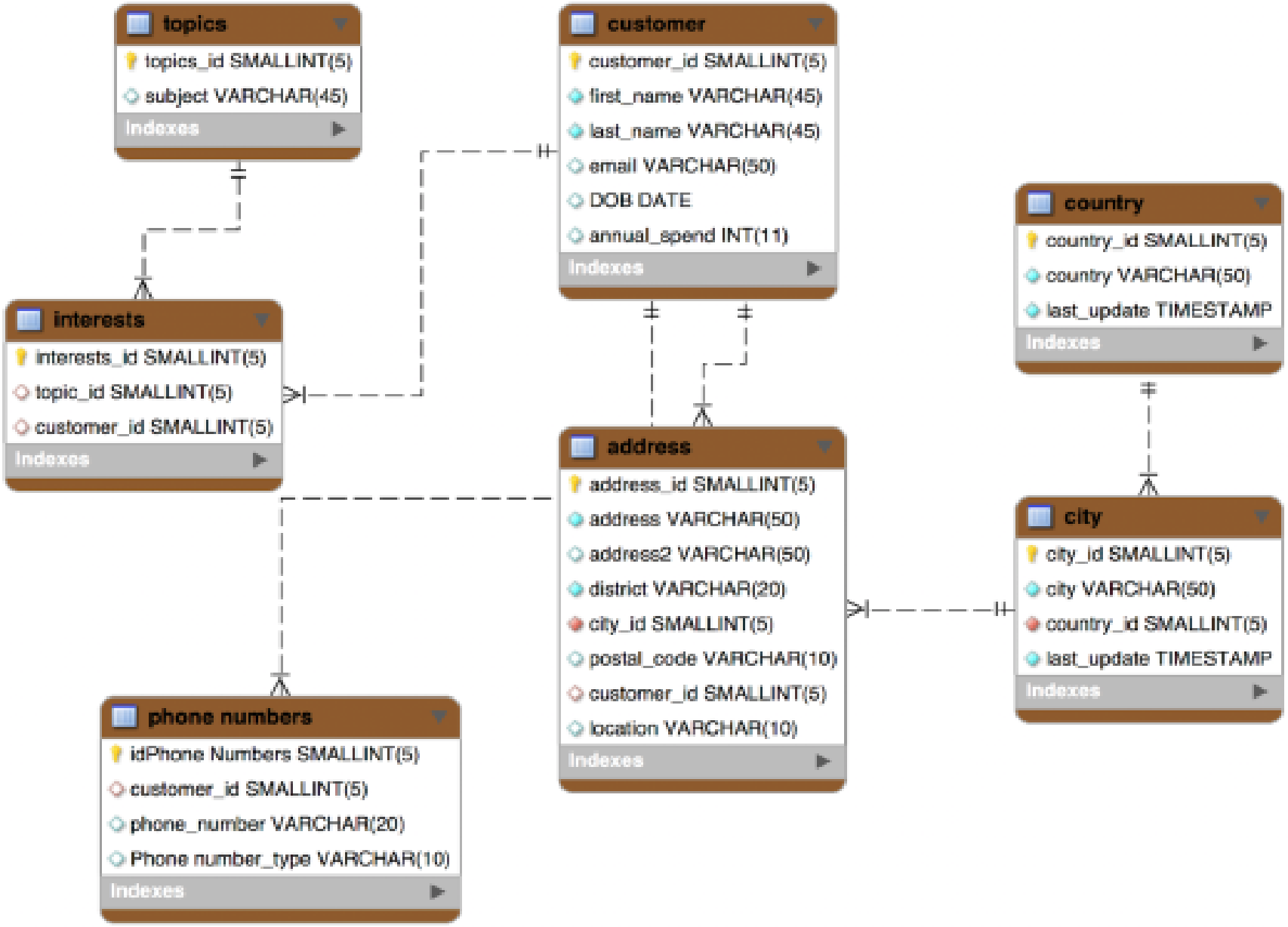
Embedding is Denormalization. It promotes data redundancy.

Referencing is Normalization. It avoids data redundancy.

High-Level Modeling Difference

- Data that belongs to parent-child relationships in relational tables would commonly be collapsed (embedded) into a single document in document database
- Relational database avoids redundant data
- Document database tends to promote redundant data

Relational Data Model - Customer



Document Database Counterpart


```
{
"_id":
ObjectId("5ad88534e3632e1a35a58d00"),
"name": {
"first": "John",
"last": "Doe" },
"address": [
{ "location": "work",
"address": {
"street": "16 Hatfields",
"city": "London",
"postal_code": "SE1 8DJ"},
"geo": { "type": "Point", "coord": [
51.5065752,-0.109081]}},
+ {...}
],
"phone": [
{ "location": "work",
"number": "+44-1234567890"},
+ {...}
],
"dob": ISODate("1977-04-01T05:00:00Z"),
"retirement_fund":
NumberDecimal("1292815.75")
}
```

End Results

- Complete document can be accessed with a single call to database, rather than having to JOIN multiple tables to respond to a query.
- A document is physically stored as a single object, requiring only a single read from memory or disk.
- Relational JOINs require multiple reads from multiple physical locations.
- Avoiding redundant data helps maintain data integrity

Promoting data redundancy may improve query performance
But may cause data integrity issues