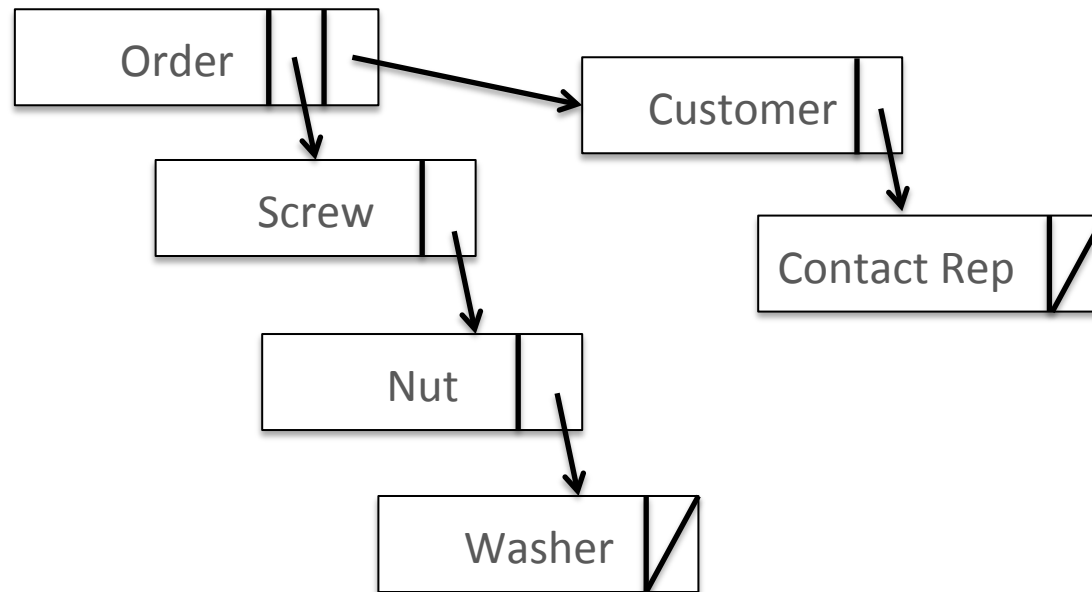


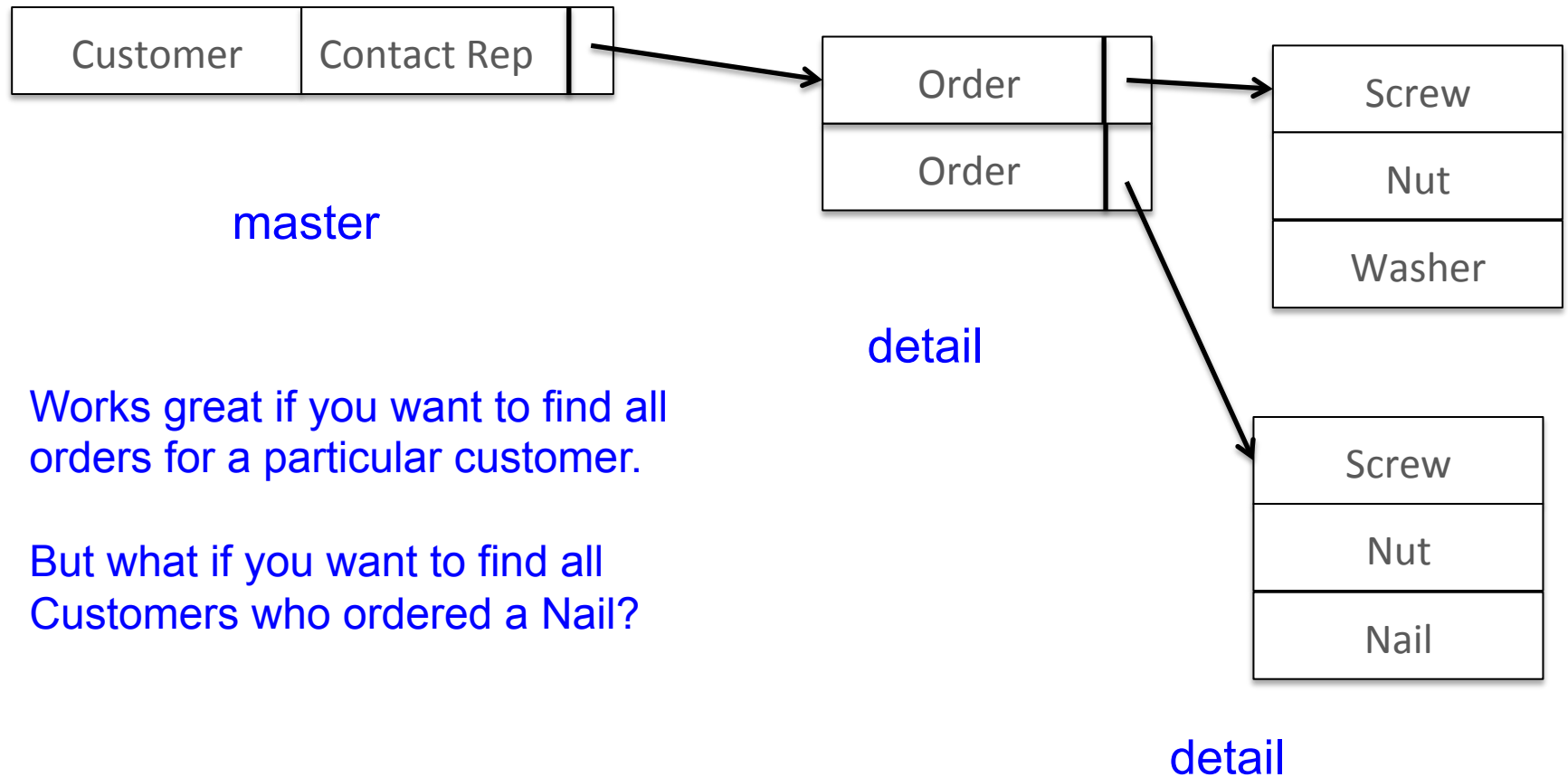
## Questions to consider

- How is the data physically organized on disk?
- What kinds of queries are efficiently supported by this organization, and what kinds are not?
- How hard is it update the data, or add new data?
- What happens when I encounter new queries that I didn't anticipate? Do I reorganize the data? How hard is that?

## Historical Example: Network Databases



# Historical Example: Hierarchical Databases



# One view

“Relational Database Management Systems were invented to let you use one set of data in multiple ways, including ways that are unforeseen at the time the database is built and the 1st applications are written.”

(Curt Monash, analyst/blogger)

## *Relational* Databases (Codd 1970)

- Everything is a table
- Every row in a table has the same columns
- Relationships are implicit: no pointers

Course	Student Id
CSE 344	223...
CSE 344	244...
CSE 514	255..
CSE 514	244...

Student Id	Student Name
223...	Jane
244...	Joe
255..	Susan

## *Database Philosophy*

God made the integers;  
all else is the work of man.

(Leopold Kronecker, 19<sup>th</sup> Century Mathematician)

Codd made relations;  
all else is the work of man.

(Raghu Ramakrishnan, DB text book author)

# Relational Database History

Pre-Relational: if your data changed, your application broke.

Early RDBMS were buggy and slow (and often reviled), but required only 5% of the application code.

*“Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed.” -- Codd 1979*

**Key Ideas:** Programs that manipulate tabular data exhibit an algebraic structure allowing reasoning and manipulation independently of physical data representation