1., Currently, the user relation is not normalized, for there is no superkey, but the second relation is because of pId being a suoerkey and cId being an attribute in the candidate key.

Given:

Email -> name

Date, email, pId -> price, amt

--------------

R1 = (Email, name)

R2 = (Date, email, pId, price, amt)

R3 = (Email)

2.

1. Neo4j - connecting friends would be most logically understandable as a graph database because of all the possible interleaving between friends and products.
2. MongoDB - implementing a coupon exchange system would be best done in a key-value database because keys can have pseudo random id numbers that are attached to the corresponding value (Percentage, reduction, etc...).
3. Cassandra - the analyst could quickly traverse a single column (where each row is a different year).
4. MongoDB - a key-value pair would be best and most efficient for linking a company (key) to its profile (value).
5. MongoDB - a key-value pair can represent a shopping cart effectively and efficiently.