**Data Anomalies**

Consider the following design where every data point is stuffed into one matrix. In the following:

* ST is some number that refers to the shipping status (e.g 20 means order shipped and 30 means received)
* PNO are parts
* QTY is quantity
* SNO is some shipping number (shipping ID)

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

| SNO | ST | PNO | QTY | CLR |

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

| S1 | 20 | P1 | 300 | red |

| S1 | 20 | P2 | 200 | blue |

| S1 | 20 | P2 | 400 | blue |

| S1 | 20 | P4 | 200 | gray |

| S1 | 20 | P5 | 100 | black |

| S1 | 20 | P6 | 100 | white |

| S2 | 30 | P1 | 300 | red |

| S2 | 30 | P2 | 400 | blue |

| S3 | 30 | P2 | 200 | blue |

| S4 | 20 | P2 | 200 | gray |

| S4 | 20 | P4 | 300 | gray |

| S4 | 20 | P5 | 400 | black |

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

1. What is CRUD? What commonly-used web-programming pattern supports CRUD?
2. What is insert anomaly? Show how the above design suffers from it. How can SQL solve it.
3. What is delete anomaly? Show how the above design suffers from it. How can SQL solve it.
4. What is update anomaly? Show how the above design suffers from it. How can SQL solve it
5. What is normalization?
6. What are the advantages of normalization?
7. What are the disadvantages of normalization?
8. What is nosql?
9. How does nosql address data anomalies (hint: trick question)
10. What kind of applications should use nosql. Give an example and explain.
11. What kind of applications should not use nosql? Give an example and explain.