1 a False
b True
c True
d False. This statement is true of every table.
e True
f True
g False
h False

2 a i (a,b,c)
ii Yes it is necessary. Modify to place B-R-C in aggregate where B-R is a total-participation, key relation (i.e., B=>R --- thick line and arrow) and then relate aggregate to A with a new relation
iii Yes it is necessary. Add a new entity set that has r as a key attributed and change R into a quaternary relation (i.e., 4-ary).

b
1. A->B given
2. A->A,B augmentation (of 1)
3. A,B->C given
4. A->C transitive (1 and 3)
5. A,E->C,E augmentation
6. C,E->C reflexive
7. A,E->C transitive (5 and 6)
qed

c i Because of the fact that there are multiple items per room, it is better to represent rooms and items as separate entity sets and to create a relationship set (e.g., "is store in") that relates rooms to items.
ii It is often useful to store summary information in an entity set for details stored in another entity set to which the first is related, but it is not necessary. If the summary of total room value isn't stored in the room entity set, then a query over the items entity set will be required to total the value of items in a room, whenever this total is needed. The design tradeoff is between the cost of this query, on the one hand, and the cost of maintaining the summary on every change to the item entity set on the other. The principle for deciding is to compare these costs along with the predicted relative frequency of reading room-value totals on the one hand and inserting or changing individual item values on the other. For example, if items are entered or changed rarely, but room value is examined frequently, it makes sense to store the summary in room. Otherwise, it might not make sense.

3 a sin and sname,address are the candidate keys
b I can't underline, so I'll list the primary keys in all caps.
Swimmer(SIN, sname, address, school, group)
Event(STYLE,distance)
ScheduledHeat(NUMBER, time, style)
Swam(SIN, NUMBER, result)
c Swimmer1(SIN, sname, address, group)
Group(GROUP, school)
Event(STYLE,distance)
ScheduledHeat(NUMBER, time, style)
Swam(SIN, NUMBER, result)

4 a R1=(A,B)
R2=(B,C,D,E)
It in 3NF because A is a superkey to R1 and B and B,C are superkeys to R2. To see why B->BCDE:

1. B->CD given
2. B->C decomposition (of 1)
3. B->BC augmentation (of 2 with B)
4. BC->E given
5. B->E transitive (of 3 and 4)
6. B->CDE union (1 and 5)
7. B->B reflexive
8. B->BCDE union (6 and 7)
qed

It preserves the loss-less join property, because R1 intersect R2 is B and B is a key to R2.
It is dependency preserving, because B->CD and BC->E can be verified by R2 alone. And A->ABCDE can be simplified to A->B, because B->BCDE and thus it can be verified by R1 alone.
many other answers are okay too

b PI(a,b,c,d,e,g,h) (RHO(X.a? a, X.b?b) (SIGMA(X.a=Y.a ^ X.b=Y.b) (X x Y)
I didn't check for RHO. I only took off .5 if you missed the PI and I didn't take off even this if you indicated that the solution had only one copy of a and b.