

EX1:

MANAGER(manager-id, manager_name),

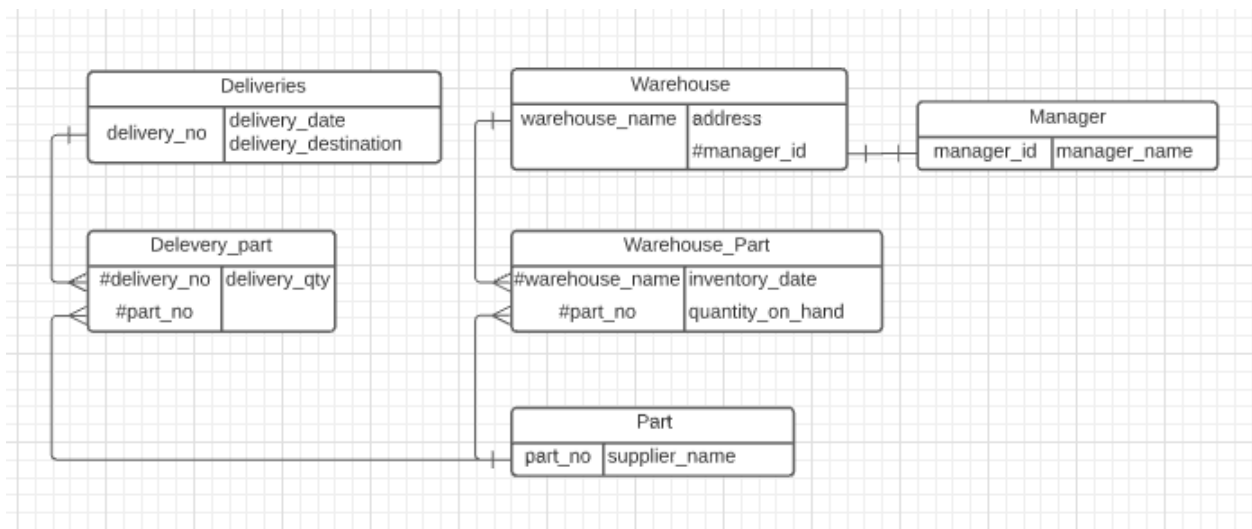
WAREHOUSE(warehouse-name, warehouse-address, manager-id#),

WAREHOUSE_DETAIL(warehouse-name#, part-no#, qty-on-hand, inventory-date),

PART(part-no, supplier-name),

DELIVERY_DETAIL(part-no#, delivery-no#, delivery-qty)

DELIVERY(delivery-no, delivery-date, delivery_destination)



I asked for an UML diagram. Here, it's not even a Chen diagram, as the multiplicities (cardinalities) are missing, and neither UML nor Chen diagrams mention foreign keys... At the exam, this would result in MINUS points.

EX2:

2.

Author(author_id, first_name, Last_name)

R2(#Title_id, #author_id)

Titles(Title_id, name, Domain_id)

Domain(Domain_id, name)

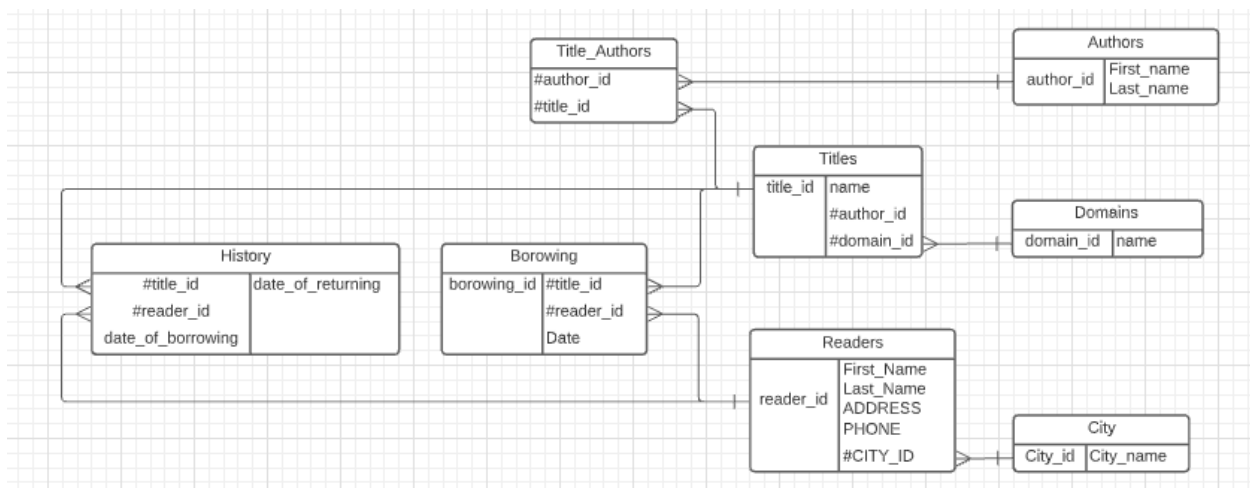
reader(reader_id, first_name, last_name, address, city_id, phone)

city(city_id, city_name)

borrowing(borrow_id, #reader_id, #title_id, date)

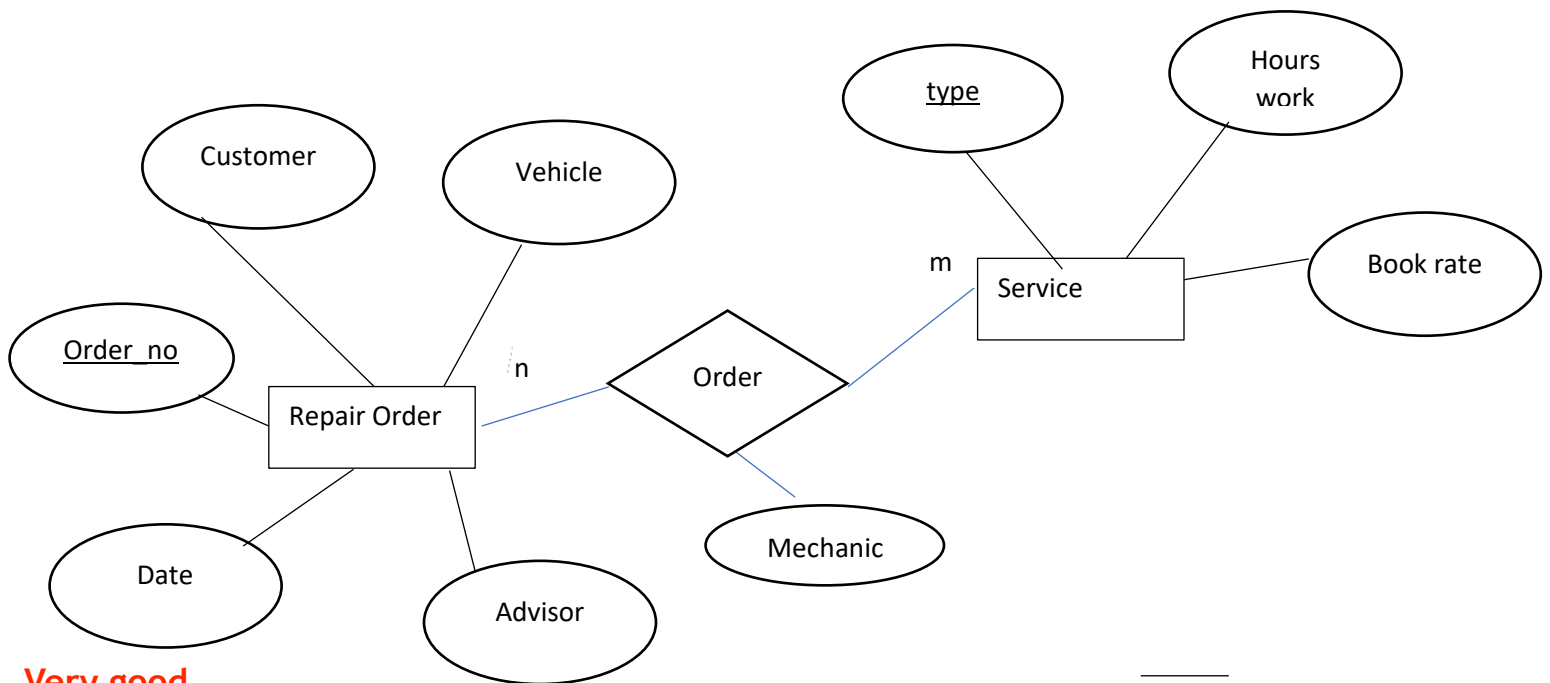
History(#borrow_id, data of borrowing, date_of_returning)

OK, you should just rename R2 for some more readable name.



You should use Chen or UML notations (show cardinalities and don't show FKs). Otherwise, at the exam, this would result in MINUS points. Also, you did not finish exercise 2!!!

Exercise 3:



Very good

2.

Repair_Order(order_no, customer, vehicle, advisor, date)

Service(type, hour_work, book_rate)

Order(order_no#, service #, mechanic)

Very good

3.

Repair_Order(order_no, vehicle#, advisor, date)

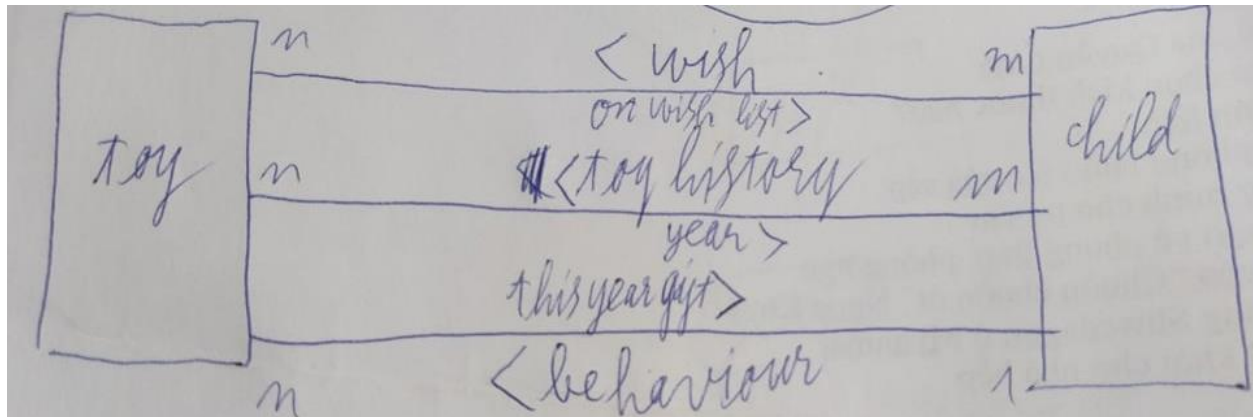
Motorbike(vehicle, customer)

Service(type, hour_work, book_rate)

Order(order_no#, service #, mechanic)

OK I see, here you decompose according to the transitive dependency vehicle -> customer. If this is the way it works in Vietnam (each vehicle belongs to at most 1 customer), then it is clever! (in France it's different; multiple people can co-own a vehicle).

Exercise 4



CHILD (child_id, name, address, birthday)

WISH(child_id#, toy_id#)

HISTORY(child_id#, toy_id#, year)

THIS_YEAR(child_id#, toy_id#, behavior)

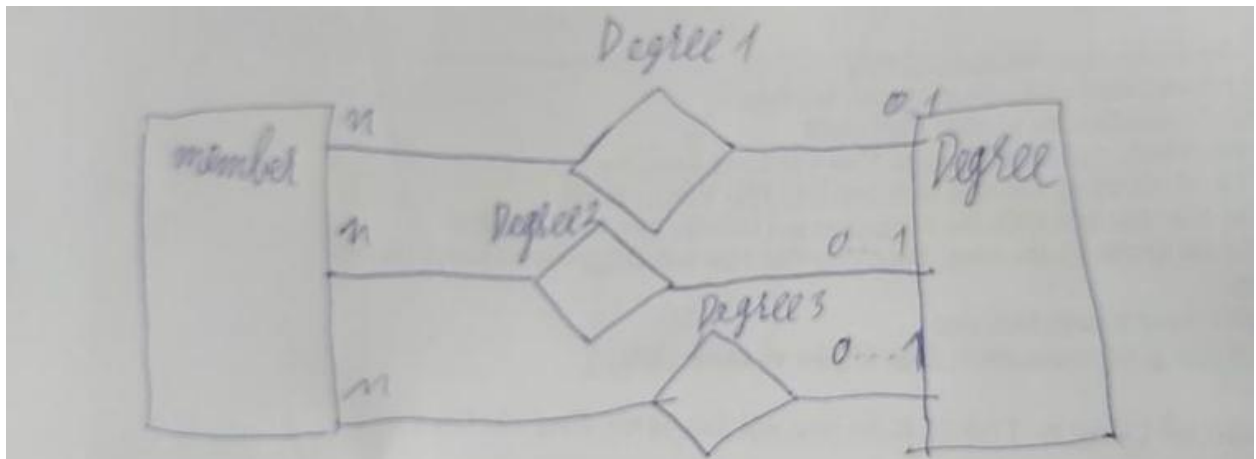
TOY(toy_id, toy_model)

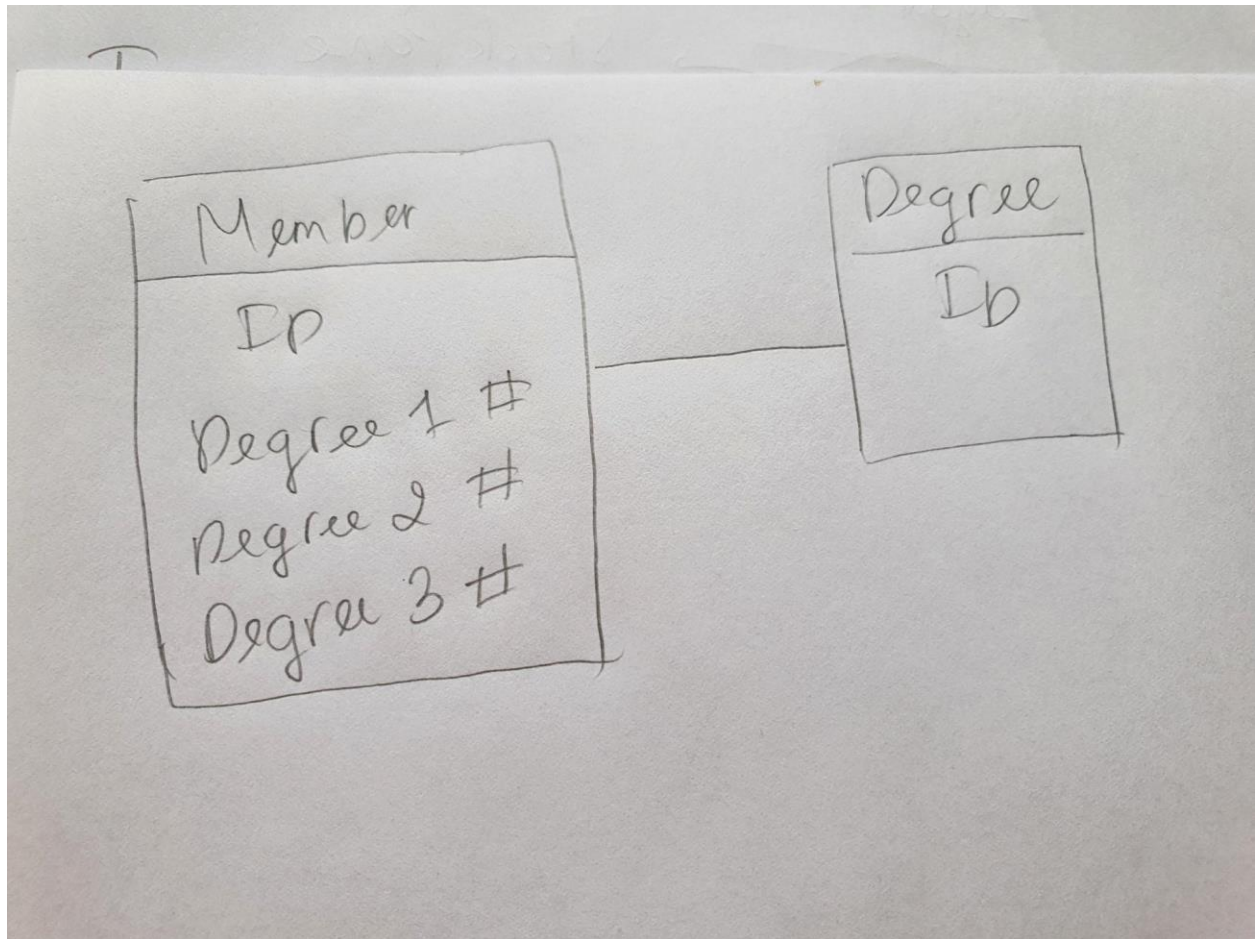
Multiple problems here:

- this year, the child might have multiple gifts (cardinality m, not 1)
- if this is the case, then there is redundancy because behaviour is repeated multiple times
=> have a look at Group 5's solution, and the one I propose, both are correct.

Exercise 5:

1. Drawback: Redundant, hard to control, handle
- 2.

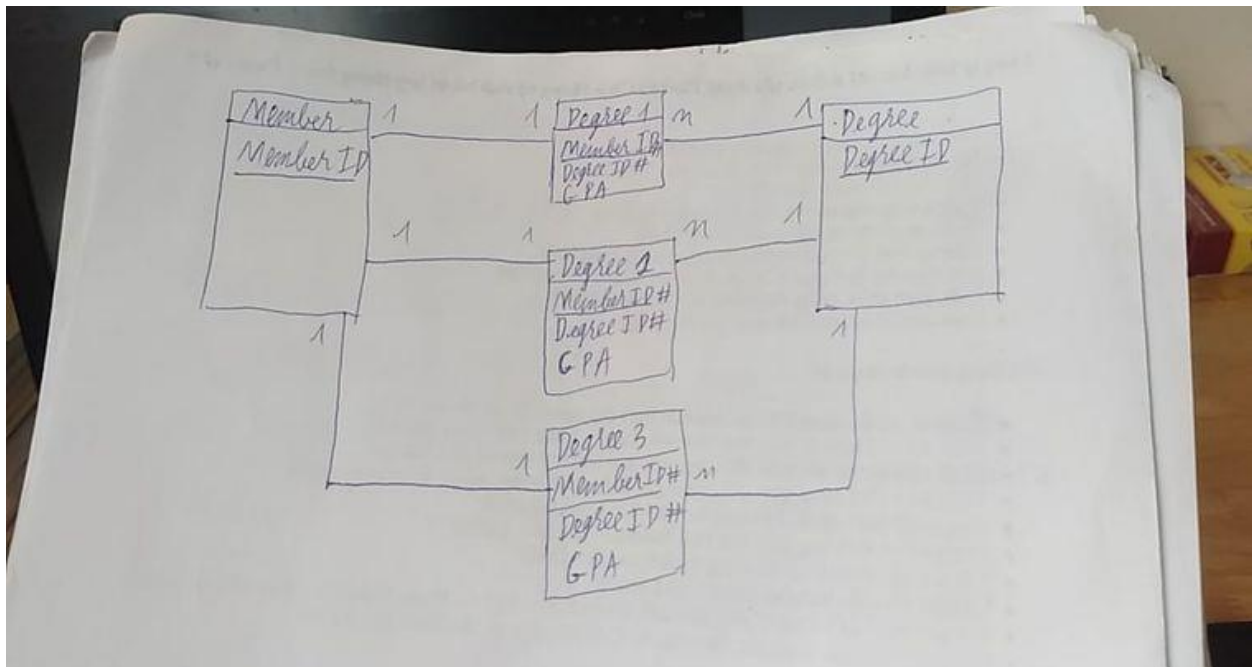




3. The later one is better.

You did not really reply to exercise 5. See the correction from Group 2

4.



5.

MEMBER(member_id, facFirstName, facLastName)

DEGREE1(member_id#, degree_id#, GPA)

DEGREE2(member_id#, degree_id#, GPA)

DEGREE3(member_id#, degree_id#, GPA)

DEGREE(degree_id, level, year, major, university)