Group 3:

- Nguyen Hoang Dang – 20194423
- Do Quoc An – 20194414
- Ha Vu Thanh Dat – 20194424
- Ta Viet Cuong – 20194422
- Pham Thanh Truong – 20194460
- Pham Nhu Thuan – 20194456
- Bui Anh Duc – 20194426
- Nguyen Van Toan – 20194457

Exericise 1:

1.

MANAGER(manager-id, manager-name, warehouse-name)

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

PARTS(part-no, supplier-name)

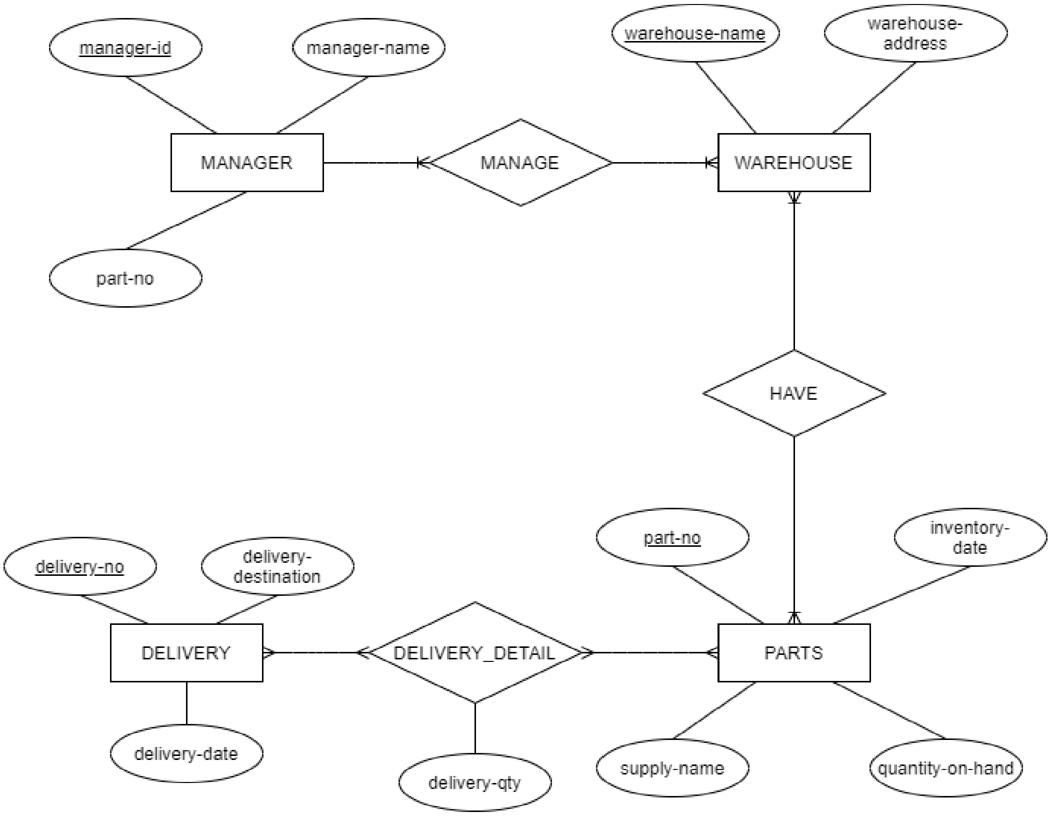
PART_DETAIL(part-no#, warehouse-name#, inventory-date, qty-on-hand)

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

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

In your schema, one warehouse is managed by only one manager, and a manager manages only one warehouse. This is not specified in the text but why not?

2.



I asked for an UML diagram. Here, it's not even a Chen diagram as the multiplicities (cardinalities) are missing...

Also:

- why is part-no an attribute of the entity MANAGER???
- the inventory_date and the quantity-on-hand depend on the warehouse, and the specific part -> they should be attributes of the relationship « HAVE », not of the entity « PARTS »

Excersise 2.

I don’t see what changes in the BORROWING table? OK for the HISTORY table.
You missed the inverted FKs between TITLES and DOMAIN…

1.

BORROWING table changes to BORROWING(borrowing_id, reader_id#, title_id#, date)

HISTORY table changes to HISTORY(borrowing_id#, date_of_borrowing, date_of_returning)

Very good, knowing that the relationship « BORROWED » will be implemented in the database as the table « HISTORY »

2.

These table are not in 3NF. Because in READERS table

Address -> city_id and city_id-> city_name is transitive dependence

OK

3.

We separate READERS table into:

READERS (reader_id, first_name, last_name, address, city_id)

CITY (city_id#, city_name)

After that, these tables in 3NF

OK

4.

AUTHOR table and TITLE table change to

AUTHOR (author_id, first_name, last_name, title_id#)

Title (title_id, name)

In this case an author can only write ONE book!!! -> you need a mumtiple-to-multiple relationship (see the answer given by Group 1 for instance)

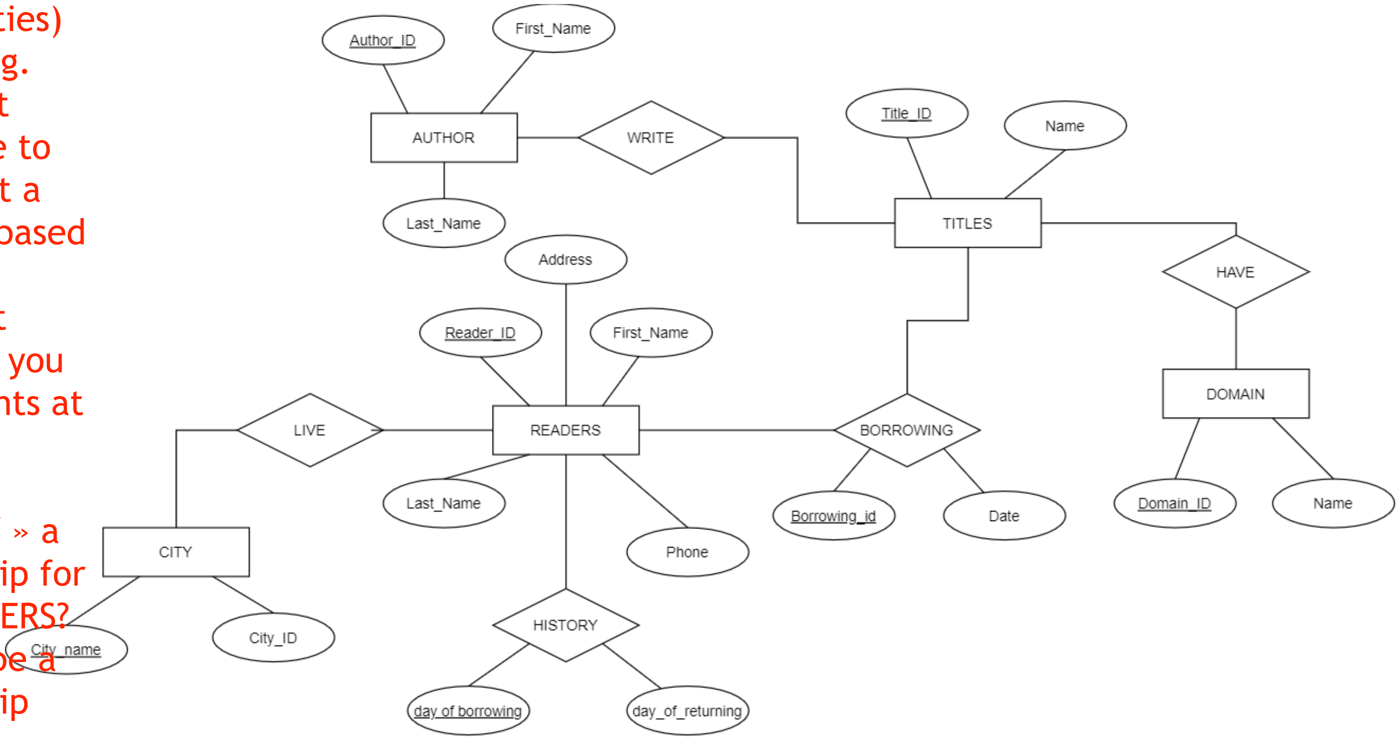
5.

In my opinion, it can be a good idea. But the idea that you have instead one single BORROWING table with a DATE_OF_RETURNING column, is a not good idea. Because when admind entry HISTORY table and want to review history, information of reader and book is necessary.

Your argument is not clear…

Once again, the multiplicities (cardinalities) are missing. It makes it impossible to implement a database based on your schema. It would get you minus points at the exam. Why is « HISTORY » a relationship for only READERS? It should be a relationship between READERS and TITLES…

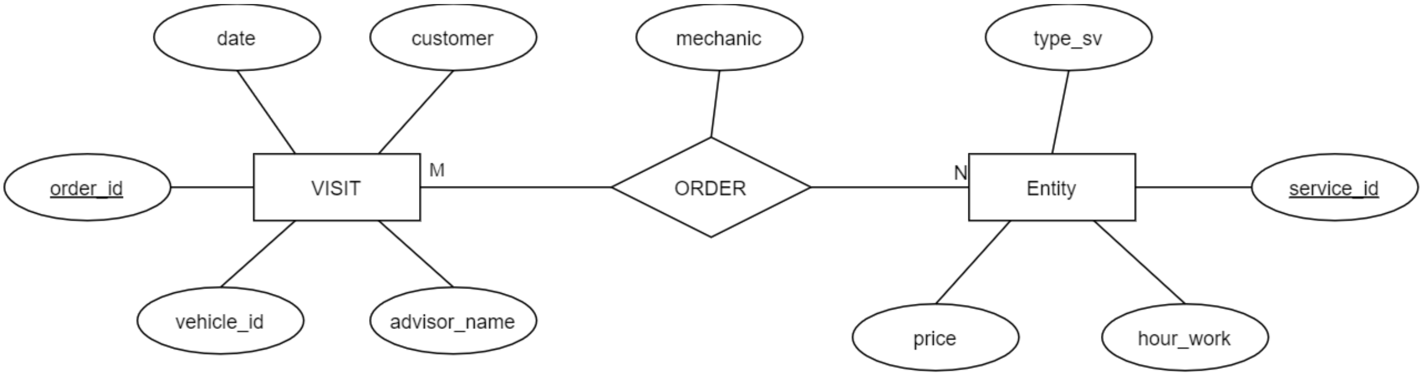
6.



Exercise 3.

1.

Good (if you replace « entity » by « service »…)



2.

Visit(order_id, customer, vehicle_id, advisor_name, date)

Service(service_id, order_id#, type_sv, price, hour_work)

Order(order_id#, service_id#, mechanic)

3.

This schema is in 3NF because all attributes are atomic, full dependent, and this schema has no transitive dependency.

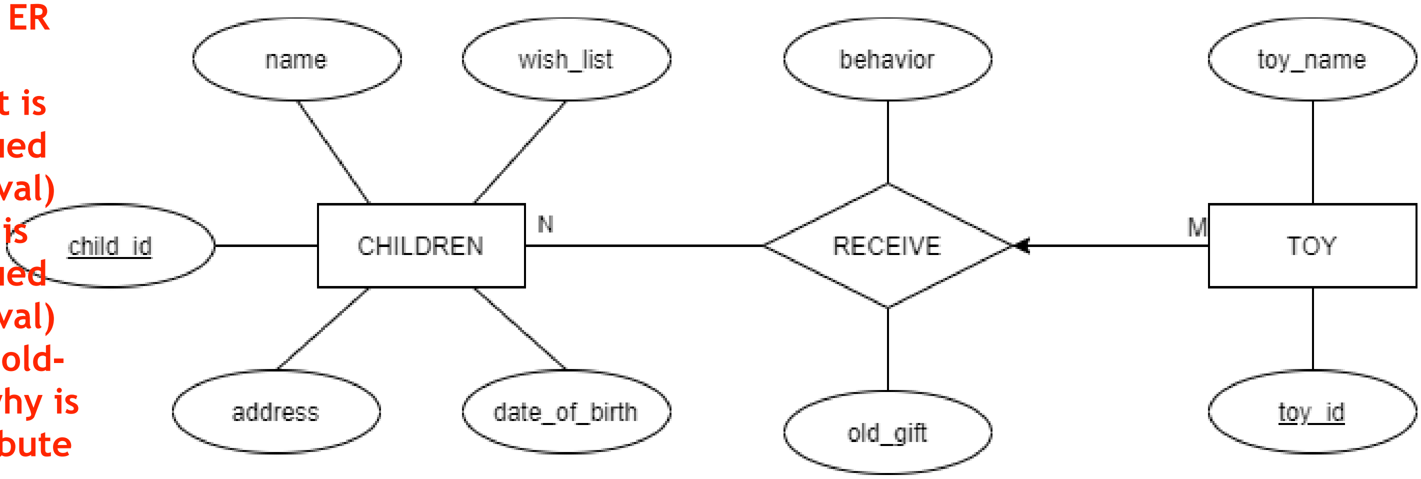
OK

Exercise 4.

Several problem with your ER diagram:
- wish-list is multi-valued (double oval)
- old-gift is multi-valued (double oval)
- what is old-gift and why is it an attribute of the relationship RECEIVE???

Have a look at Group 5's solution, and the one I propose, both are correct.

1.



2.

Children(child_id, name, address, date_of_birth, wish_list)

Toy(toy_id, toy_name)

Receiving(child_id#, toy_id#, behavior, old_gift)

This schema is not adapted to the problem (for the reasons above)

3.

This schema is in 3NF because all attributes are atomic, full dependent, and this schema has no transitive dependency.

Exercise5.

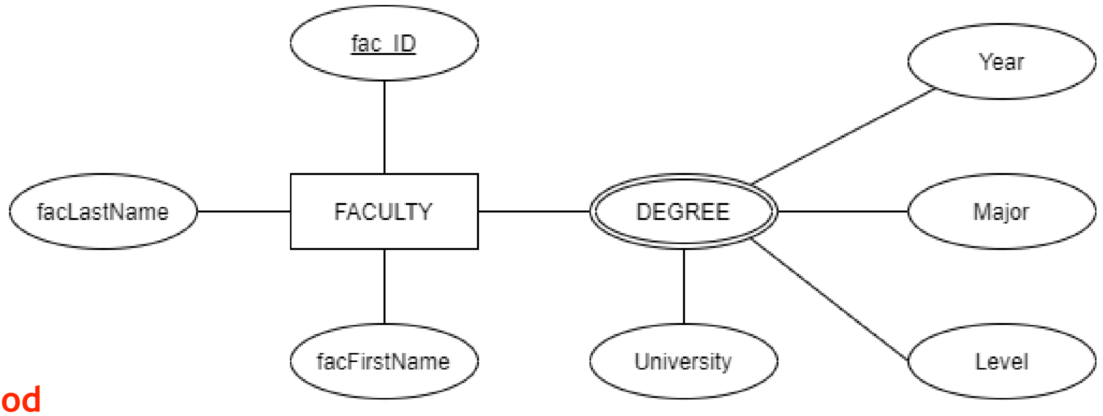
1.

Each member can have more degrees than 3 because they can have many degrees in the same level.

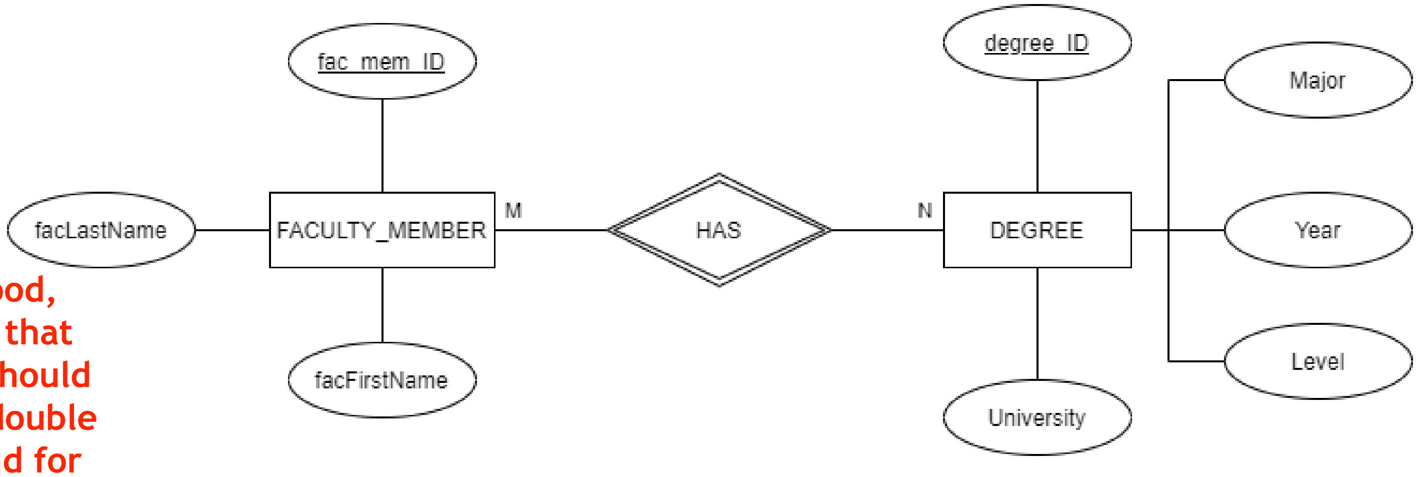
Each degree is described by the same properties (level, year, major, university) but not in the same order. When users insert records into database, the data can be mismatch.

True. And also, wasted storage space.

2.



Very good



Very good, except that there should be no double diamond for the relationship HAS

3.

In my opinion, the best diagram for store the GPA of the faculty members of each degree is the second diagram.

Very good

4.

FACULTY_MEMBER(fac_mem_id, first_name, last_name)

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

DEGREE_DETAIL(fac_mem_id#, degree_id#, GPA)

Very good

5.

This schema is in 3NF because all attributes are atomic, full dependent, and this schema has no transitive dependency