Exercise 1.

1.

MANAGER (**manager_id**, manager_name)

WAREHOUSE(**warehouse_name**, warehouse_address, manager_id#)

PART(**part_no**, inventory_date, supplier_name)
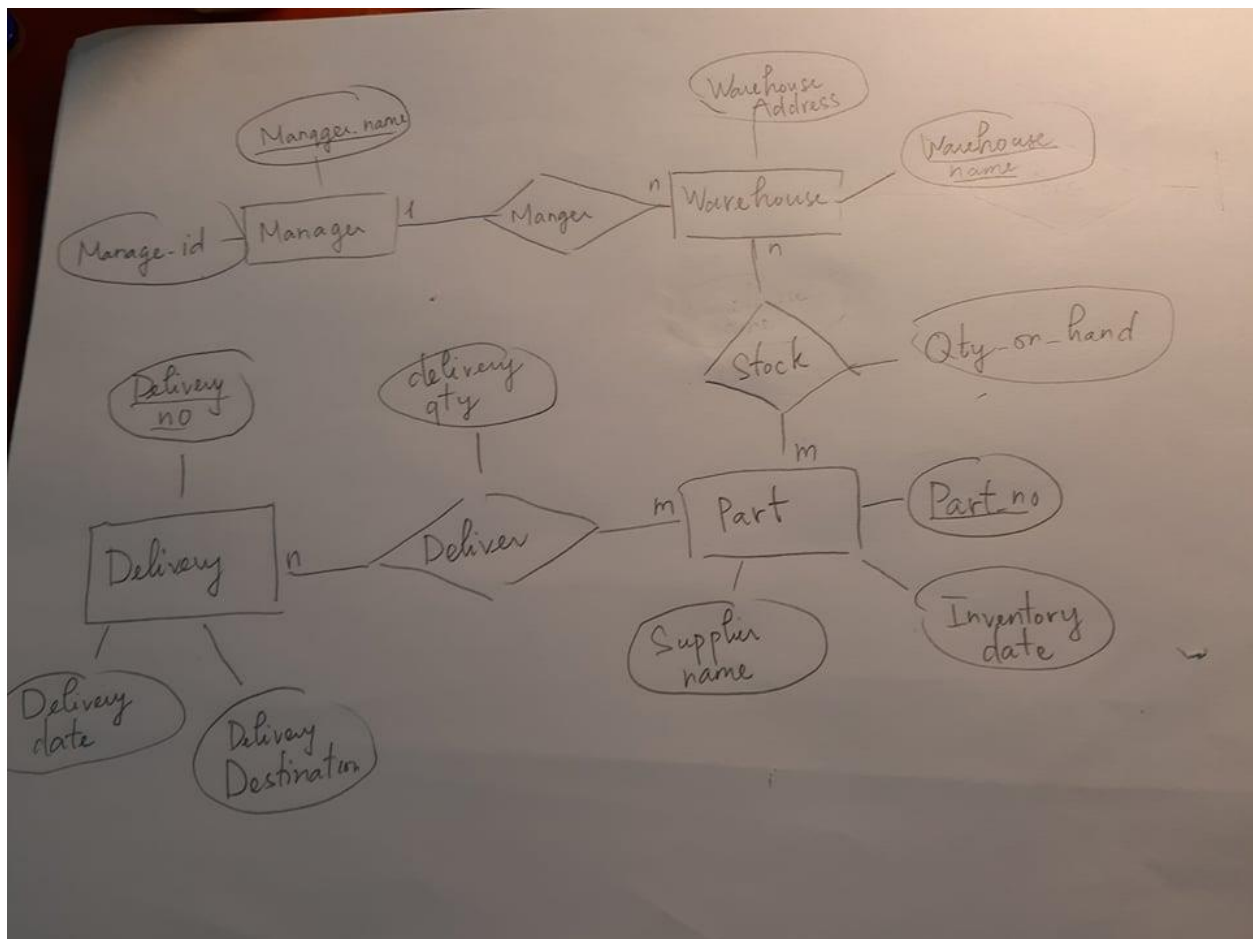
PART_QTY(**part_no#**, **warehouse**#, qty_on_hand)

DELIVERY(**delivery_no**, delivery_date, delivery_destination)

DELIVERY_PART(**delivery_no#, part_no**#, delivery_qty)

2.

Exercise 2.

1.

TITLES table

      •TITLE_ID, PK

      •NAME

      •Author_ID, FK

      •DOMAIN_ID, FK

DOMAIN table

      •DOMAIN_ID, PK

      •NAME   **OK but also HISTORY's PK is not unique**

2. It is not because CITY_NAME is functional dependent on CITY_ID

OK

3. ,

READERS table
• READER_ID, PK
• First_Name
• Last_Name
• ADDRESS
• CITY_ID, FK
• PHONE

CITY table

• CITY_ID, PK
• CITY_NAME

OK

4.

We can not store that information in the above database. We can remove the Author_ID from the TITLES table and create a new table: WRITES (**Author_ID#**, **Title_ID#**)

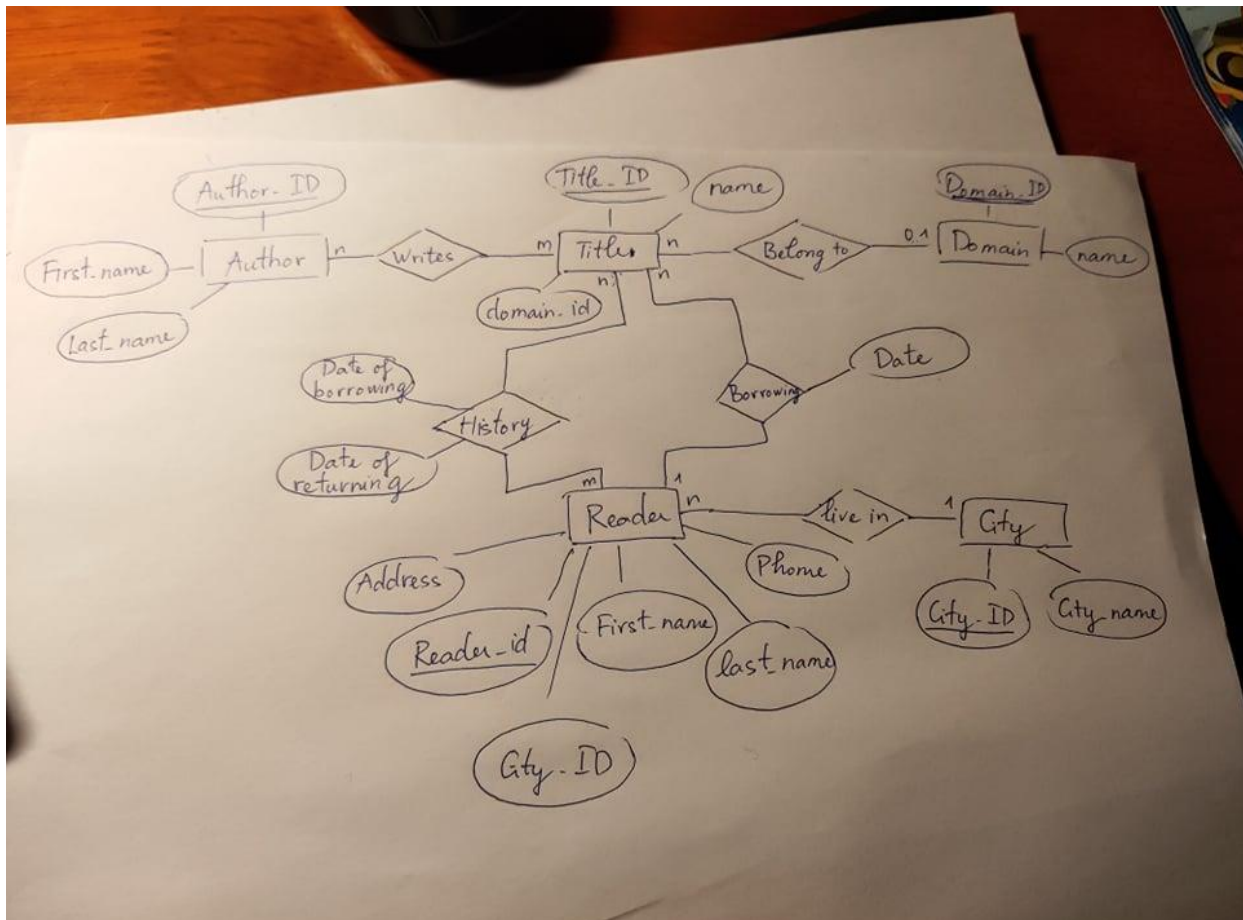**Very good**

5.

This is a good idea to separate the books which are being borrow and the books have been returned.
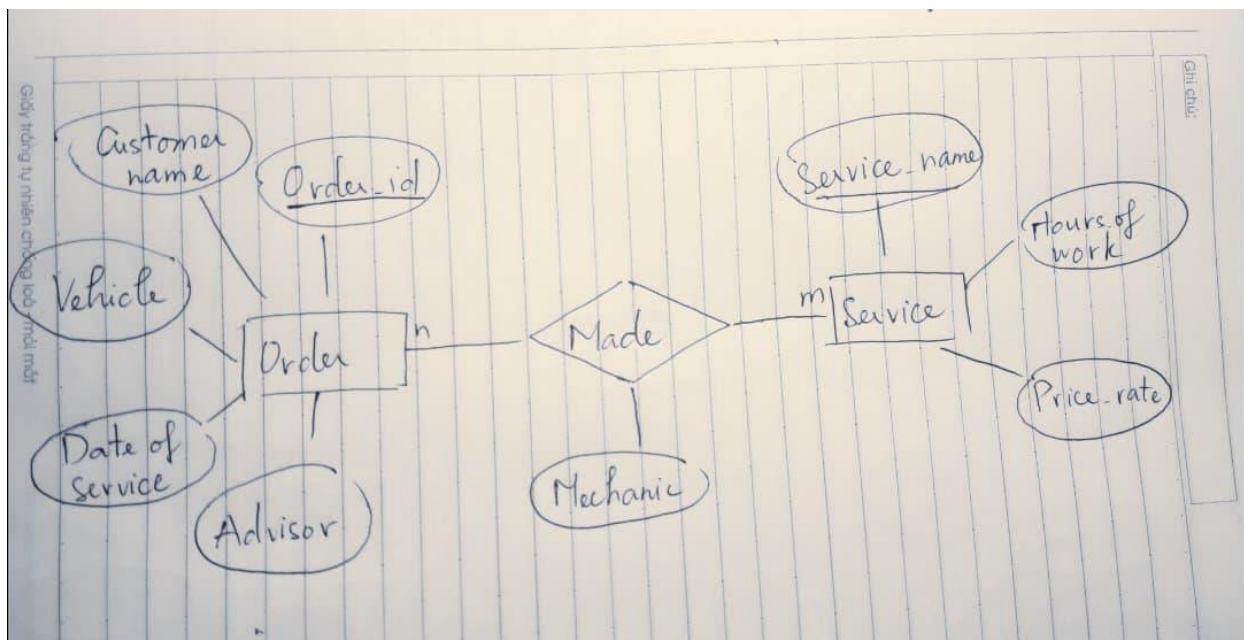
OK

6.

OK but there is a problem in your cardinality for the BORROWING relationship (should be n-m). Also in Chen notation we only put the max cardinality (here 1, not 0..1), and we don't put the FKs (here domain_id as an attribute of Title)

Exercise 3:

1.

Very good

2.

ORDERS (**_order_id_**, customer_name, vehicle, date_of_service, advisors)

SERVICES (**_service_name_**, hours_of_work, price_rate)

MADE (**_order_id_**#, **_service_name_**#, mechanic)

Very good

3.

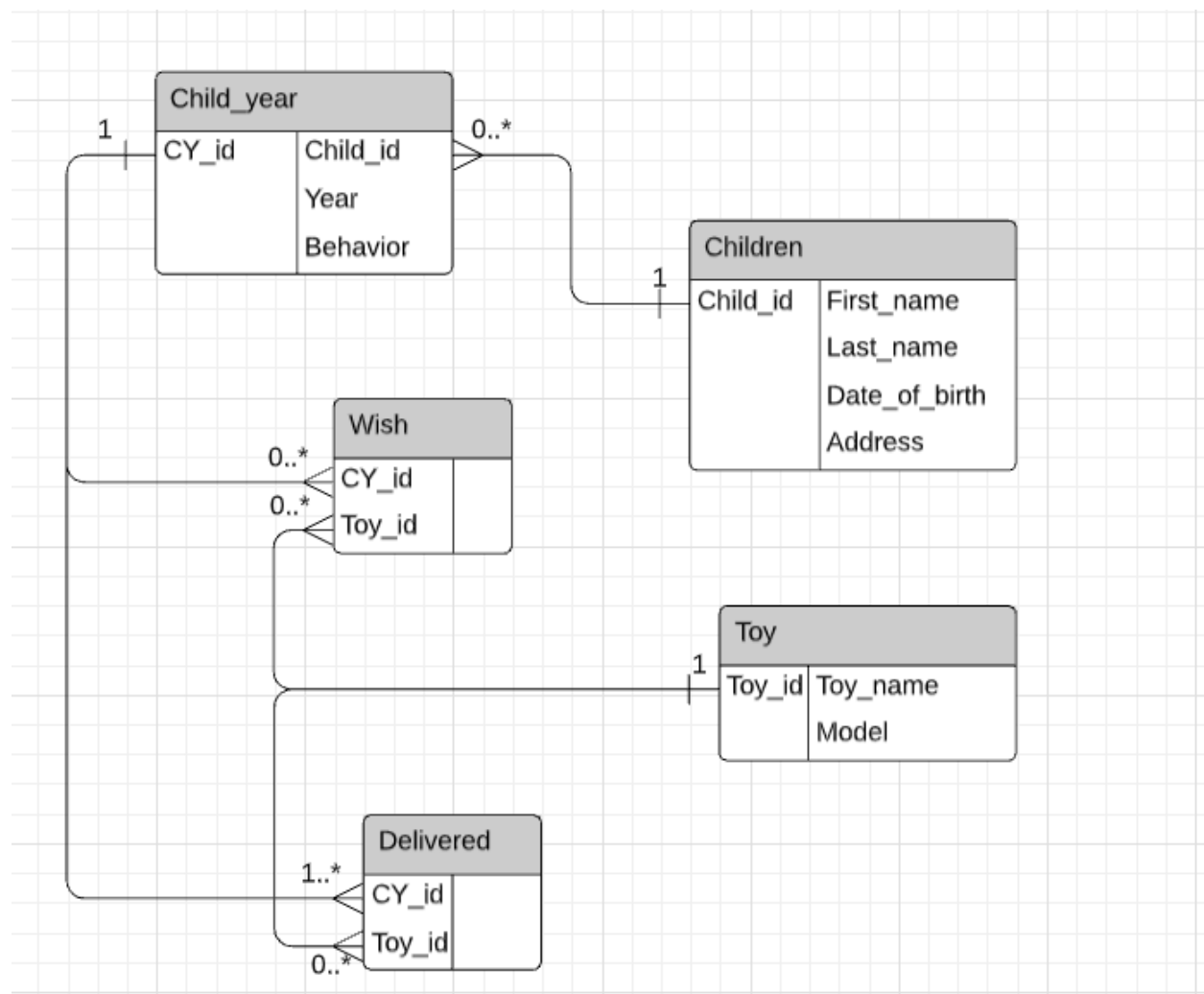The scheme is in 3NF.

OK

Exercise 4.

1.

Very good model. Just have a look at the alternative model that I propose (unifying « wish » and « delivered » using Boolean attributes).

2.

Children(**Child_id**, First_name, Last_name, Date_of_birth, Address)

Children_year(**CY_id**, Child_id, Year, Behavior)

Wish(**CY_id#, Toy_id#**)

Delivered(**CY_id#, Toy_id#**)
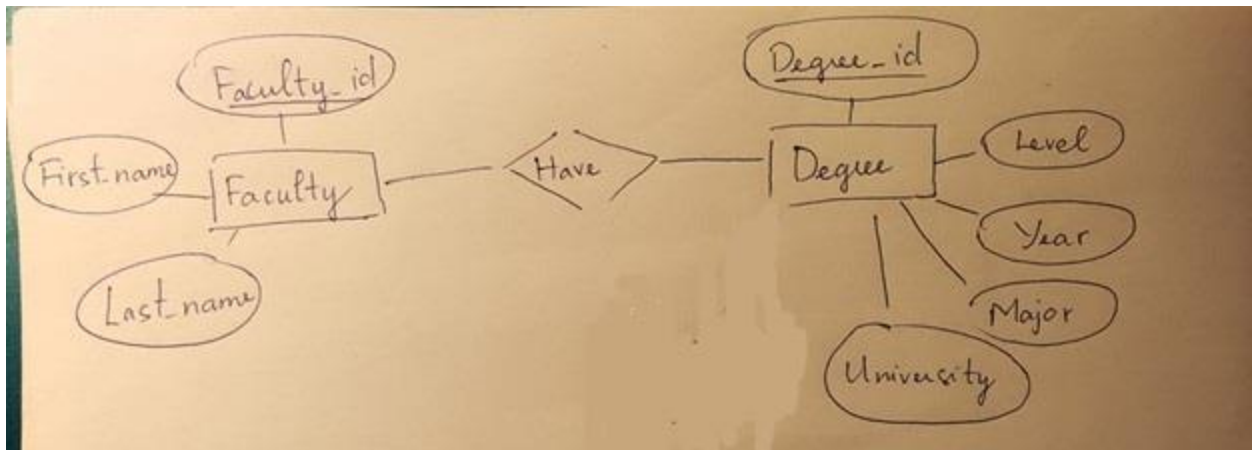
Toy(**Toy_id**, Toy_name, Model)

3

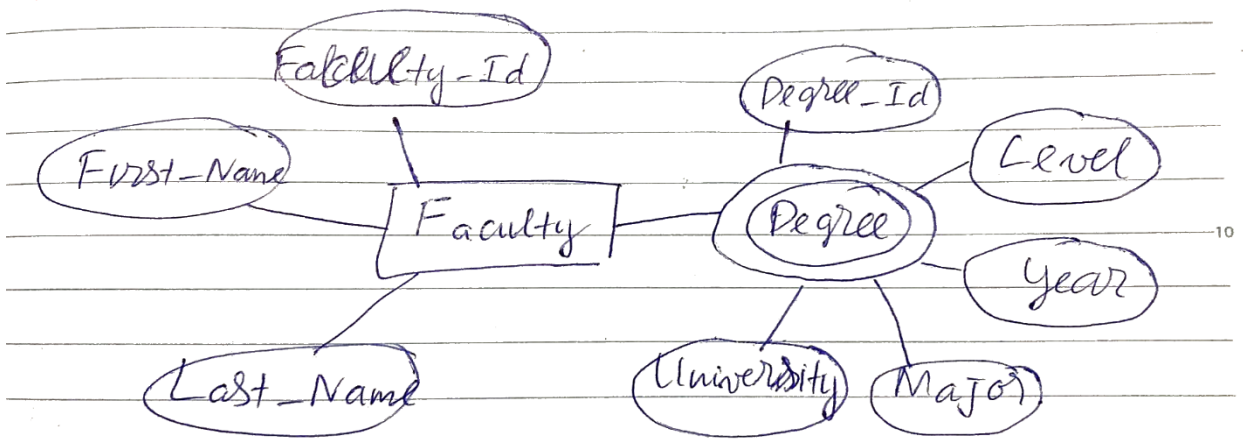The relational schema is in 3NF

Very good

Exercise 5.

1. Too many null value, hard to count/group similar degrees

2.

Very good, but don't forget the multiplicities (here n-m) : otherwise you'll have MINUS points and at the exam.

Very good

3.

The first ERD is best for that purpose, just add attribute *grade* to relation *HAVE.*

4.

FACULTY (**faculty_id**, first_name, last_name)

HAVE (**faculty_id #**, **degree_id#**, grade)

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

Very good

5.

The relational schema satisfy 3NF

Group members

DANG THANH LAM 20194442
LE HAI SON 20194449
NGUYEN DUY HUNG 20194436
NGUYEN HOANG NHAT QUANG 20194448
NGUYEN QUANG TUNG 20194462
DANG VU HOANG HIEP 20194431
VU QUOC HUY 20194439
LAI QUANG HUY 20194438