Model Paper

Section A

① b/c
② c
③ c
④ a
⑤ e

⑥ c
⑦ b
⑧ A
⑨ A
⑩ D

⑪ c
⑫ a
⑬ b
⑭ c.
⑮ b

⑯ b
⑰ a
⑱ c
⑲ a
⑳ a

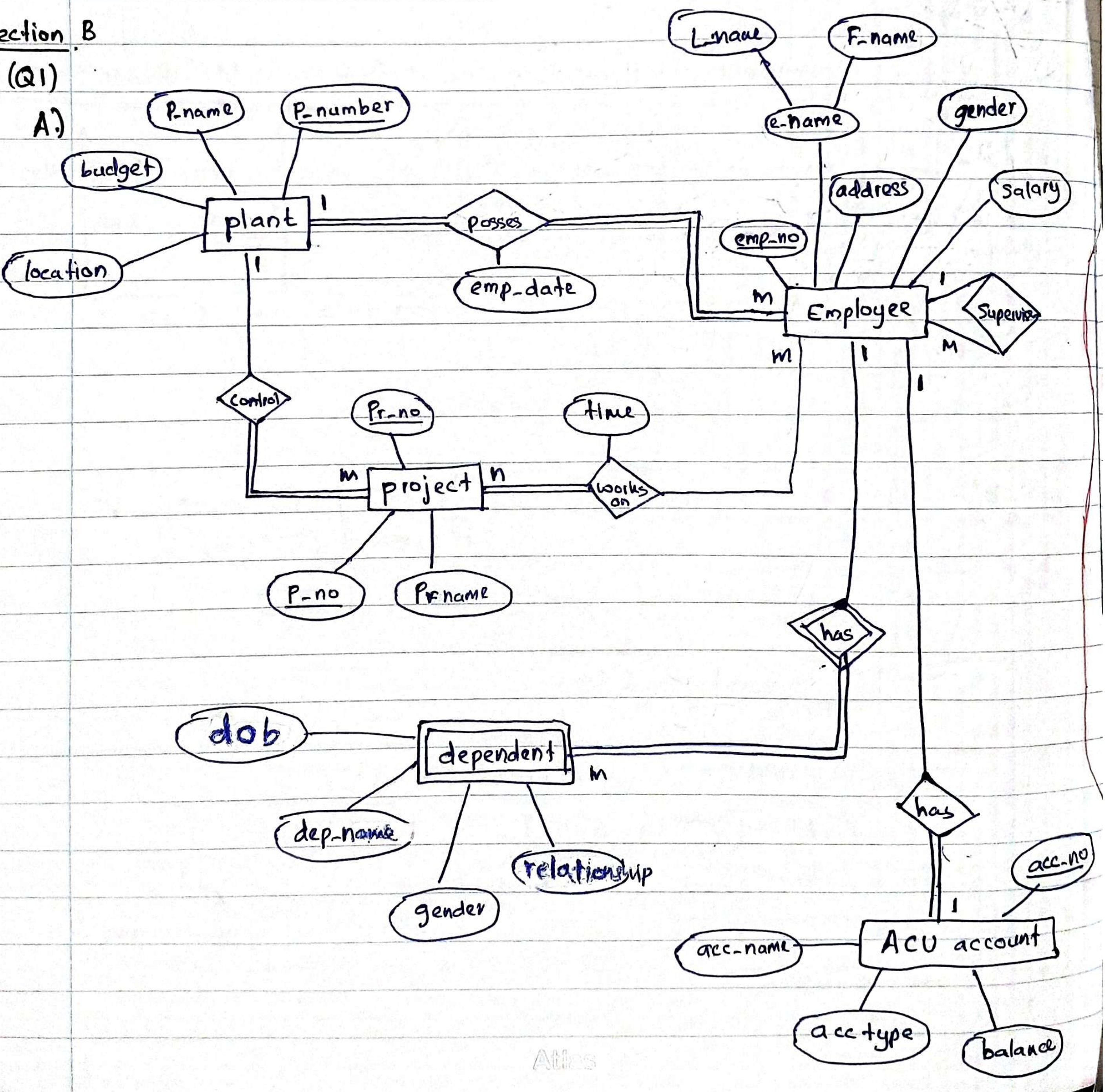
+ ⑨ The first normal form is used to eliminate duplicate info.

* ⑫ % is used with like and _ is used to fill the character

Section B

(Q1)

A)



No : Date: / /

Assumptions.

1.) Every plant must have atleast one employer,

2.) I assume that emp-date attribute. is for plant and employee

3.) I assume that employee name is a composite key as First name and Last name

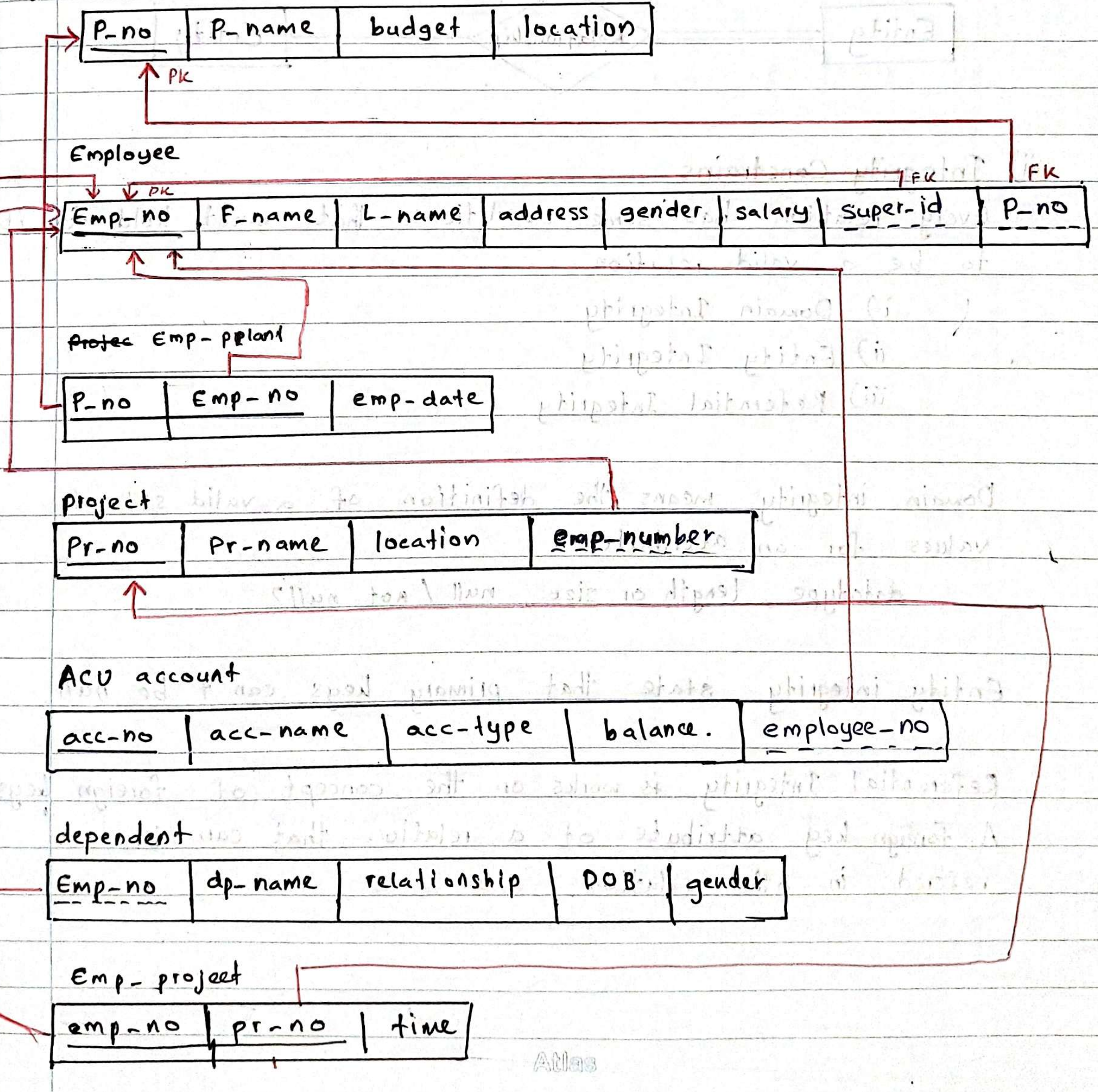
4.) I assume that one employee can have only one ACU account.

5.) Every plant do not need to work on a project

6.) Every employee do not need to work on a project.

7.) Every employee must work for a project;

B.)



Plant

P_no	P_name	budget	location

Employee

Emp-no	F_name	L-name	address	gender	salary	super-id	P_no

~~Projec~~ Emp-pplant

P_no	Emp-no	emp-date

project

Pr-no	Pr-name	location	emp-number

ACU account

acc-no	acc-name	acc-type	balance.	employee-no

dependent

Emp-no	dp-name	relationship	DOB	gender

Emp-project

emp-no	pr-no	time

No : Date: / /

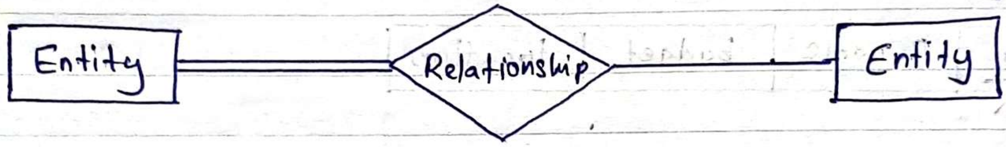
(Q2) A)

i) Participation ~~constraints~~ constraints

Specify whether the existance of an entity depends on its being related to another entity via the relationship type

i.) Total participation - Each entity is involved in the relationship. It is represented by double lines.

ii) Partial participation - Not all entities are involved in the relationship. Partial participation is represented by single lines.



ii) Integrity Constrains

Every relation has some conditions that must hold for it to be a valid relation.

i.) Domain Integrity
ii) Entity Integrity
iii) Referntial Integrity

Domain integrity, means the definition of a valid set of values for an attribute.

datatype, length or size, null / not null?

Entity integrity, state that primary keys can't be null.

Referential Integrity works on the concept of foreign keys. A foreign key attribute of a relation that can be refered in other relation.

B.)

A	B	C	D	E	F	G	H	I	J

1NF - FFD

A	B	E	F	H

~~2NF~~

A	D	J

B	C	G

2NF - PD → If a primary attribute is there it should be a 2NF (PD)

G	I

3NF - TD → No primary key attribute.

c) pets [pet-id , pet-name , pet-type , pet-age]

procedures [proc-no , proc - name]

owner [owner , visit-date , pet-id]

Pets

Pet_id	Pet-name	pet-type	pet-age
246	Rover	Dog	12
341	Jerry	Cat	4
519	Tweddy	Bird	2

Procedures

Proc-no	Proc-name
01	rabies vaccination
05	heart worm test
10	examine and treat
12	eye wash
20	annual check up

Owner

Owner	Visit-date	Pet-id
Sam	Jan 13/2014	246
Sam	Jan 23/2014	341
Cook	Mar 27/2014	246
Cook	Jan 13/2015	341
Cook	Apr 02/2015	246
Terry	Apr 30/2014	519
~~Tweddy~~		
Kim	Apr 30/2015	519

Q4.)

i.)
```
Create  table  itemCategory (
    item Category  char (1) not null,
    CategoryName   varchar (50),
    Primary key (itemCategory),
    check (itemCategory  In ('s', 'M', 'c', 'D'))
) ENGINE = INNODB;
```

```
Create  table   stock (
        itemCode   varchar (10),
        item Category   char (1)  not null,
        item Description   varchar (255)  not  null,
        QuantityInHand   int,
        CostPrice   int,
        SalesPrice  int,
        Primary key  (itemCode),
        Foreign key  (itemCategory)  References  itemCategory (itemCategory),
        Check (costprice >100   and  costprice <10000)
) ENGINE = INNODB;
```

ii)
```
Insert  into   stock   values (1015, 'M', 'LED Monitor', 14,
10500, 12500);
```

iii)
```
select  *  item Description , item Category
from  stock
where  item Category  = 'D'
ORDER By   item Description ;
```

iv.)
```
select   itemcode , item Description
From   stock
where   Quantity In Hand  < 50 ;
```

v.)
```
update stock
    set costprice = costprice * 1.05 , salesprice = salesprice * 1.06
    where itemCategory = 'M';
```

vi.)
```
select  s.itemcode , i.category , s.costprice , s.salesPrice
from  stock as s
Join  itemcategory as i
on  s.itemcategory = i.ItemCategory
where  itemCategory  Like  '%abc%' ;
```

vii)
```
select  sum(costprice)  as  total_ costprice
from  stock;
```

viii) ~~Delete~~
```
Select  itemCategory ,  AVG(costprice)  AS  Averagecost
from  stock
Group by (itemcategoiy);
```

ix)
```
Delete  from  stock
where  itemcode = '1012' ;
```