Ques 5).

Ans 2 NF (second normal form).

ISBN → Title
ISBN → Publisher
ISBN → publisher & publisher → Address } = ~~ISBN → Address~~ ISBN → Address

Thus ISBN is a primary key

As ~~ISBN~~ title, publisher, address ~~has~~ are functionally dependent on ISBN, Thus, they are not multivalued attributes Nor Composite ⇒ obeys 1NF.

As there is only ~~primary key~~ attribute in the primary key it obeys 2NF { because all ~~the~~ non prime attribute depend on ~~the~~ all attributes of primary key

ISBN → Address is transitive dependency because ISBN → publisher, publisher → Address ~~de~~ functional dependencies exist.

Thus dont obey 3NF.

so Ans 2NF (second normal form)

Ques 4) ~~a) i) Customer Name~~ a.city Salesman ~~b.city~~

a) i)

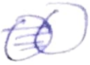
Customer Name	a.city	Salesman	b.city	b commission
Julian Green	London	Nail Knite	Paris	0.13
Graham Zusi	California	Nail Knite	Paris	0.13
Jozy Altidor	Moscow	Paul Adam	Rome	0.13

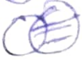
Note A INNER JOIN B on <conditions> : where A, B relation

As a.city <> b.city so customer with ~~id~~ customer-id. 3002, 3007, 3004, 3001 aren't considered.

customer with customer.id = 3009. His salesman-id is 5003. so salesman name is Lauson Hen with commision = 0.12 } in question asked for >0.12

)

) ii). NOTE :- A left ~~inner~~ outer join B on condition -  where A, B are relations. (

A right outer join B - 
on < conditions >

~~a-cust-name~~ ~~a-city~~ ~~a-grade~~ ~~Salesm~~an c-~~ord~~-no 0

cust-name	a.city	a.grade	Salesman	c.ord-no	c.ord-date	c.purch-amt
ick Rimando	New York	100	James Hoog	70013	2022-04-25	3045.6
ick Rimando	New York	100	James Hoog	70008	2022-09-10	5760
rad Davis	New York	200	James Hoog	70005	2022-07-27	2400.6
ulian Green	London	300	Nail Knite	NULL	NULL	NULL
raham Zusi	California	200	Nail Knite	NULL	NULL	NULL
eoff Cameron	Berlin	100	Lauson Hen	70003	2022-10-10	2480.4
abian Johnson	Paris	~~Nte Lu~~ 300	MC Lyon	NULL	NULL	NULL
NULL	NULL	NULL	Pit Alex	NULL	NULL	NULL
ozy Altidor	Moscow	200	Paul Adam	NULL	NULL	NULL

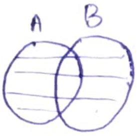
t Alex has no customer with grade not equal to NULL
ut included because right join was ~~[illegible]~~ used on customer, salesman

) → only customers with ~~id~~ customer-id = 3002, 3007, 3009 have orders with price ≯ 2000 but other customer name, city are also mentioned because left join was used

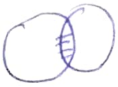
only customer with customer-id 3001, 3008 have a.city as London but for customer-id 3001 all orders have NULL values in them for a.cust_name a.city (as a.grade is NOT NULL)

so only the order by customer with customer-id 3008 will have location as London.

Thus Ans is 1 tuple. 3

NOTE :- FULL outer join :-  where A, B are relations.

Ques 2

NOTE A INNER JOIN B on <condition> :-  where A, B are relations

emp - department . DPT - NAME
IT
Finance
HR

3

there are ↓ 5,3,3,2,0 ~~0~~ employees in IT, Finance, HR, RD, QC respectively.

because there are 5 tuples in emp-details with EMP-DEPT = 57 where DPT-NAME = IT for DPT-code = 57 (Given join condition on EMP-DEPT = DPT-code). IIly others.

Ques 3

a)

AVG (pro - price)	company. mast. com_name
5000.00	samsung
650.00	iBall
1475.00	Epsion
3200.00	Asus.
500.00	Frontech.

3

~~Prints some error because the syntax of given SQL command is wrong.~~

NOTE :- A INNER JOIN B on <condition> :-

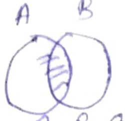
where A, B are relations.

The ~~avg~~ AVG (pro - price) = 250 for zebronics because there is only one item with pro-id = 103 ~~103~~ is ~~bought~~ ~~by~~ whose cost (pro-price) is 250.00

so that com-name not printed.

(b)

NOTE:- A INNER JOIN B on <conditions> :



where A, B are relations.

A.pro_name	A.pro_price	F.com_name
Monitor	5000.00	Samsung

0

~~Select~~ MAX(A.pro_price) gives 5000.00 and there is only one record with A.pro_price = 5000

Ques 7

~~Attempt~~

a) → it was asked to print the corresponding student stipen records.

Select *. from StudentDetails D left outer join StudentStipend S on D.StudId = S.StudId;

3

explanation :- left outer join used because we need to return student details even if stipend record is not there for that student

⇒ all the tuples in left side must ~~not~~ be printed.

so left outer join

D.StudId = S.StudId used because we need to print stipend records of that student only.

~~case 2~~ ~~If it was just asked to print student, stipend records~~ ~~D~~ ~~left~~ ~~cross join~~

~~Select * from StudentDetails~~ ~~StudentStipend~~

* → used to print all columns

b) Select distinct ~~(Stud~~ D.StudId, Name, EnrollmentNO, DateOfJoining from StudentDetails D, StudentStipend S where D.StudID = S.StudID;

3

explanation :- a student can have two ~~sti~~ stipend tuples so distinct is used so that the student information is printed only once.

D.StudID = ~~A~~ S.Stud ID condition is used to check if that student has a record or not in StudentStipend relation.

~~assuming b, c are [illegible]~~

(c) select * from StudentDetails where EnrollmentNO ~~is~~ IS NOT NULL ; (3)

[If in the question it was asked as a continuation of (b) (assuming this clause also was mentioned in (c))

select distinct D.StudId , Name, EnrollmentNO, DateOfJoining from StudentDetails D , StudentStipend S where D.StudID = S.StudID AND EnrollmentNO IS NOT NULL;].

explanation :- As he asked for those who have enrollmentNO ⇒ ~~ENRO~~ EnrollmentNO is NOT NULL

↗ must be used for NULL, NOT NULL.

(d) select ~~COUNT(*)~~ COUNT(distinct ~~stud~~ studID) as CNT from StudentStipend group by project order ~~by~~ by ~~count~~ COUNT(*) DESC; (3)

(or)

~~Create table~~ with newtable as (select count(distinct studID) as CNT from StudentStipend group by Project) select CNT from newtable order by CNT DESC;

explanation :- ~~create a~~ consider a table with CNT as the only column where it has values of ~~no~~ number of people working in a project so used Group by project.

Now from that i am ordering the values in descending order.

count(*) used because we need to calculate count of student ↳ so distinct studID used.

(e)

assuming ① n <= No of tuples in studentstipend Table

assuming ② If stipend = 3000, 1000, 1000 then 1000 is the ans for 2nd highest and 3rd highest

~~with~~
select stipend from Studentstipend where stipend IN
(select stipend from studentstipend order by stipend DESC(limit n)
order by stipend ASC limit 1;

3

explanation =

first i am selecting the n highest stipends using (select stipend from studentstipend order by stipend DESC limit n) in descending order ~~[illegible]~~ want the last record only! so, Now in this i am ordering it in ~~ascending~~ ASC (i.e ascending) and taking the 1st one i.e taking the nth highest.

Ques 6

yes! All data models should be normalised to (3NF) because there can be redundant information if not normalized to 3NF.

0.5

eg - $t_1 \rightarrow t_4$ here t_1 is primary key
$t_4 \rightarrow t_5$ t_4 is non prime attribute

<u>t_1</u>	t_4	t_5

✓ t_4 can have ~~multiple~~ same values for different t_1. Now we t_5 full functionally depend on t_4 so the same value of t_5 must be given all the same values of t_4 [redundant] information

i.e

t_1	t_4	t_5
A	Z	1
B	Z	1

=

t_1	t_4
A	Z
B	Z

t_4	t_5
Z	1

this is better!

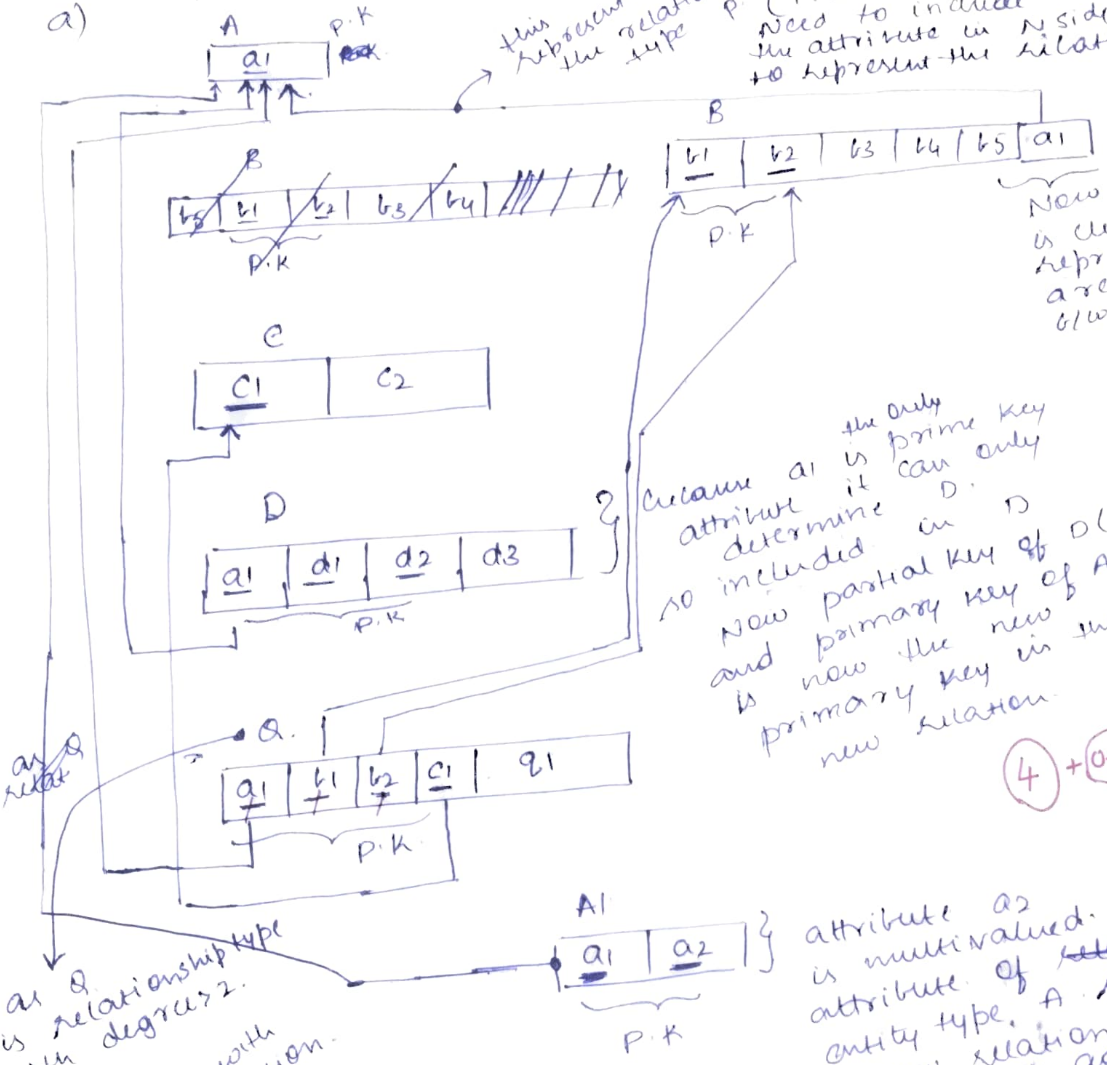
we can also have update anamolies like → insert anamolies
→ delete anamolies
→ ~~update~~ modify anamolies

Thus all data models be normalized to 3NF.

Ques 1

P.K - primary key

a)



this representing the relationship type P. (1:N relation) Need to include the attribute in N side to represent the relation.

Now it is clearly representing a relation b/w A, B

because a1 is the only prime key attribute it can only determine D. so included in D Now partial key of D(d1, d2) and primary key of A(a1) is now the new primary key in the new relation

4 + 0.5

~~as Q relat~~

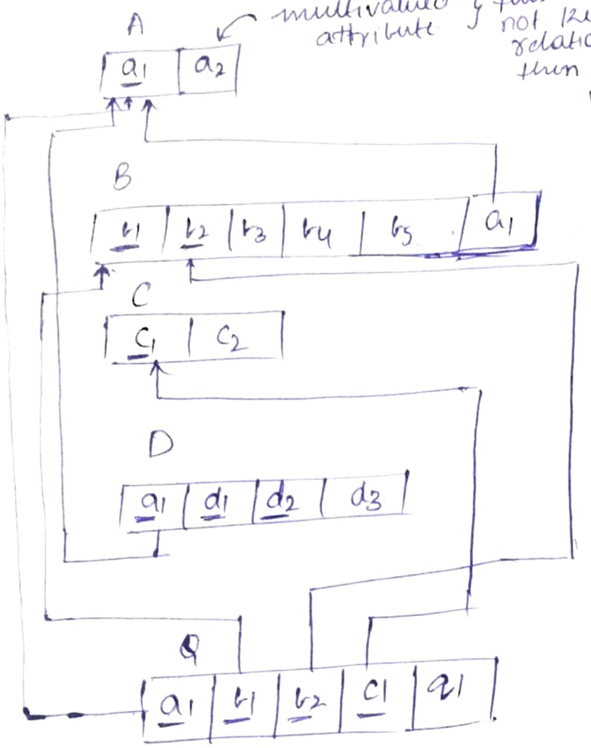
as Q is relationship type with degree>2. so it must be represented with ~~as~~ a new relation with primary key as combination of A, B, C primary keys ~~of~~

attribute a2 is multivalued attribute of ~~relatio~~ entity type. A so a new relation A1 is added for it with a1, a2 as combination is the primary key

This is the relational model!

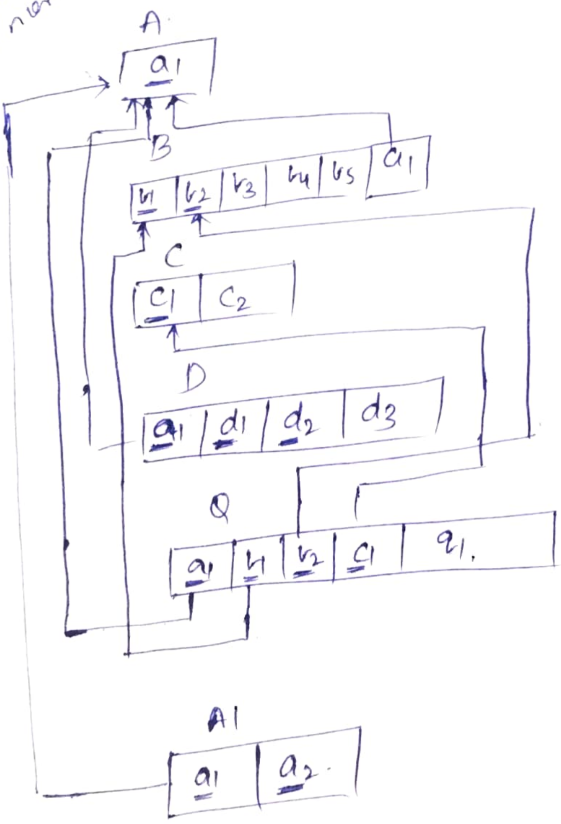
① i) unnormalised.

multivalued attribute } this is not kept as seperate relation because if is there then relational model will be in 1NF.



ii). 1NF
↳ It dont accept multivalued attributes, composite attributes, nested queries.

after 1NF normalisation.



②

iii) 2NF

$b_1 \rightarrow b_4$.

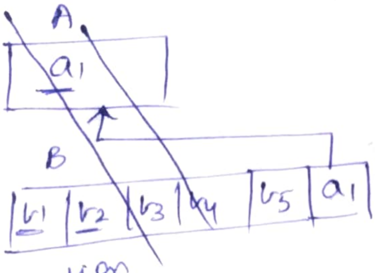
so b_4 full functionally dependent on b_1

⇒ b_4 partially dependend on $\{b_1, b_2\}$

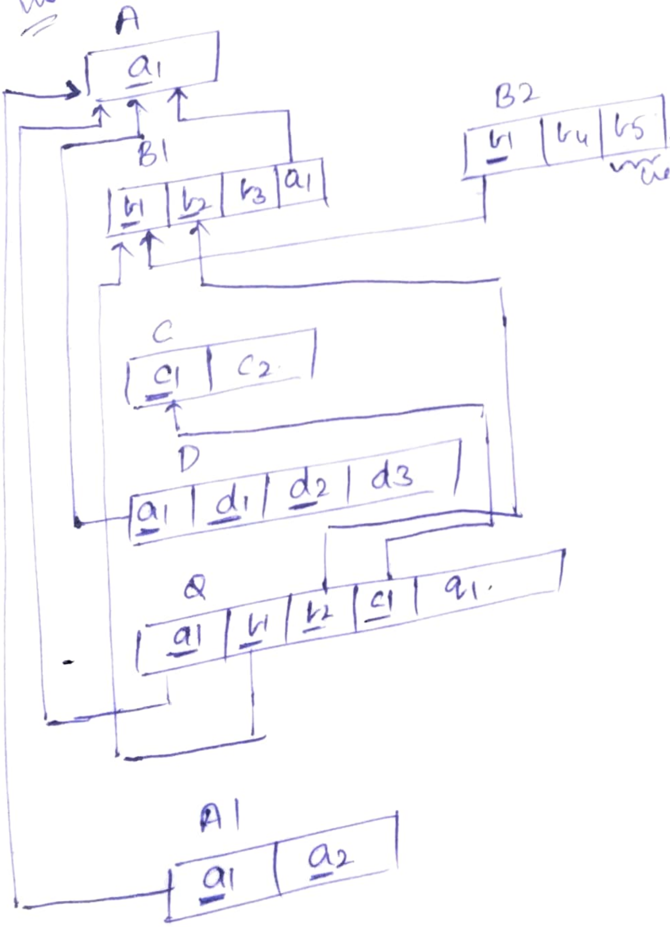
primary key

above relational model disobey 2NF.

so ⊗ ~~2NF~~



after 2NF normalization



because $b_4 \rightarrow b_5$.

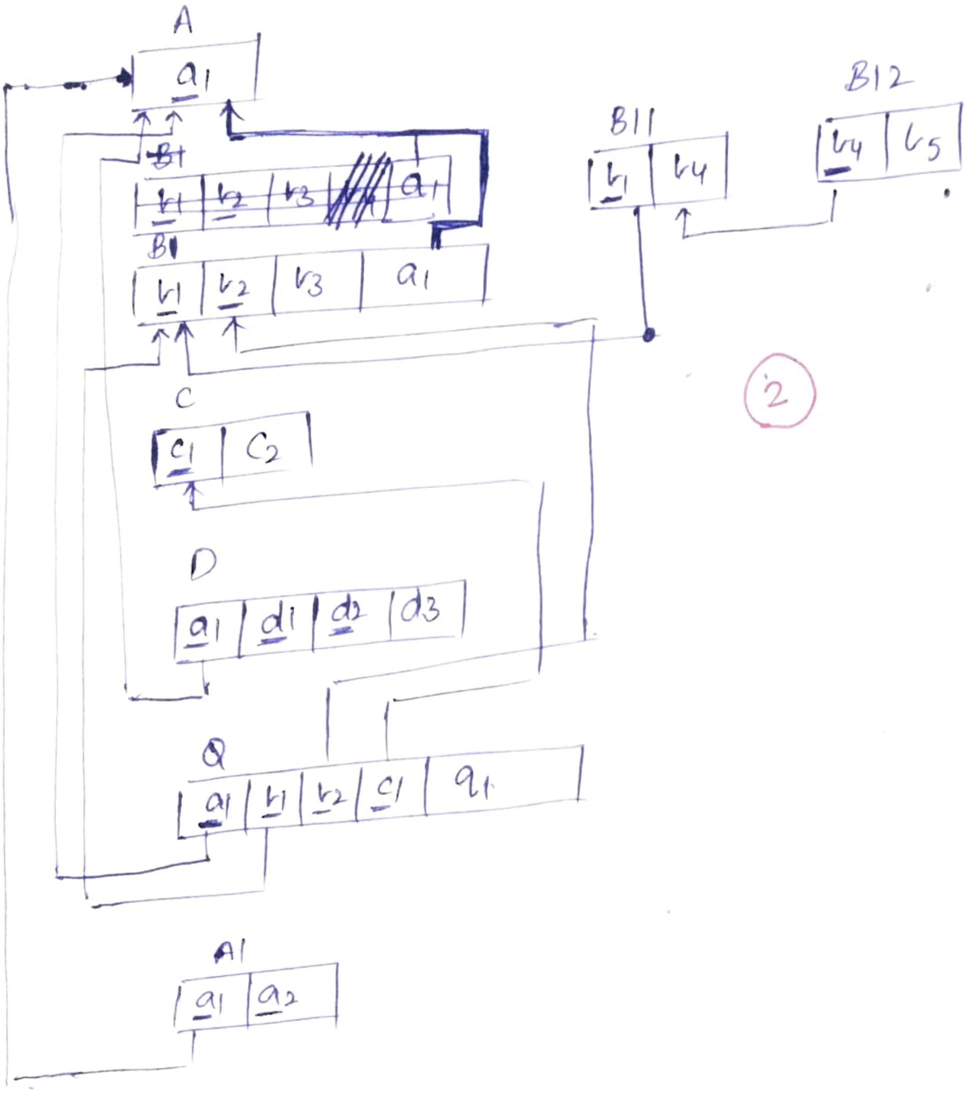
2

iv) 3NF

$b_1 \to b_5$ is transitive dependent because of $b_1 \to b_4$ & $b_4 \to b_5$

so 3NF not obeyed in the above relational model

after 3NF normalisation


A
a1
~~B1~~
~~b1 | b2 | b3 | a1~~
B1
b1 | b2 | b3 | a1
B11
b1 | b4
B12
b4 | b5
2
C
c1 | c2
D
a1 | d1 | d2 | d3
Q
a1 | b1 | b2 | c1 | q1
A1
a1 | a2