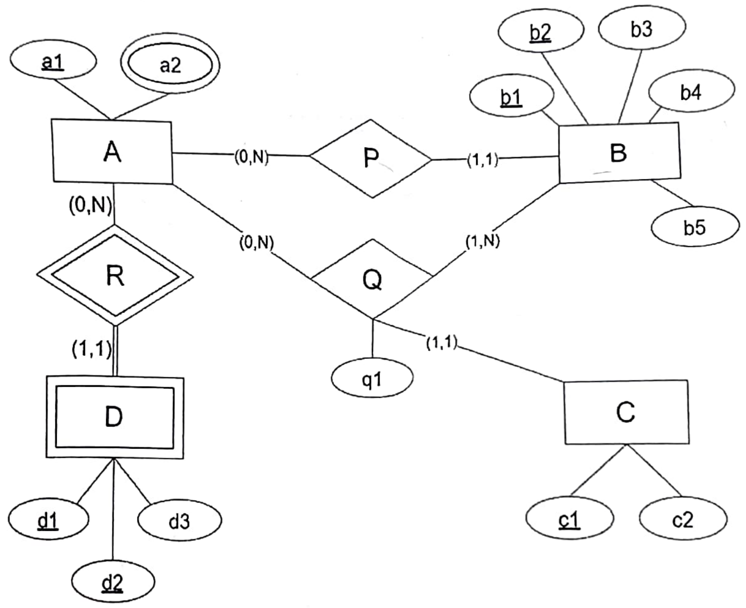
CS4.301: Data and Applications (Monsoon 2022)

End-Semester

Date: Nov 21, 2022
Time: 3 hours
Maximum Marks: 50

Ques 1. Consider the following ER diagram with the following functional dependencies.

- $b1 \rightarrow b4$
- $b4 \rightarrow b5$
- All other functional dependencies are apparent from the ER diagram
  - Each of the non-prime attributes of an entity are dependent on all of its prime attributes.
  - Each of the attributes of a relationship are dependent on the prime attributes of the participating entities.



(a) Convert the ER diagram into a relational model.

(b) Convert the resulting relational model into 1NF, 2NF, and 3NF.

Note: Multiple normal forms can be the same as each other or the same as the initial relational model.
You are expected to draw at least 1 and at most 4 relational models corresponding to each of the forms of the relational model:
(i) Un-normalized (ii) 1NF (iii) 2NF (iv) 3NF

(5+6)

Ques 2. Consider two tables namely, *emp_department* and emp_details. DPT_CODE and EMP_IDNO are the Primary Keys for *emp_department* and *emp_details* respectively. EMP_DEPT in *emp_details* is a Foreign Key referencing DPT_CODE of *emp_department.*

What will be the output for the following query?

```
SELECT emp_department.DPT_NAME FROM emp_details INNER JOIN
emp_department ON EMP_DEPT = DPT_CODE GROUP BY
emp_department.DPT_NAME HAVING COUNT(*) > 2;
```

(3)

emp_department

DPT_CODE	DPT_NAME	DPT_ALLOTMENT
57	IT	65000
63	Finance	15000
47	HR	240000
27	RD	55000
89	QC	75000

emp_details

EMP_IDNO	EMP_FNAME	EMP_LNAME	EMP_DEPT
1	Madhvi	Reddy	57
2	Pria	Khanna	63
3	Sandeep	Rajput	57
4	Ashirwad	Sharma	63
5	Piyush	Khatri	47
6	Shivani	Parashar	47
7	Sreoshi	Das	57
8	Kabir	Thapar	47
9	Naina	Talwar	57
10	Avi	Malhotra	27
11	Mohan	Bhargav	63
12	Guru	Arvind	27
13	Komaram	Bheem	57

com_name
Samsung

depart
the emp
loys more
employees

Ques 3. Consider two tables *company_mast* and *item_mast* with com_id and pro_id as their Primary Keys respectively. pro_com is a Foreign Key referencing the com_id of *company_mast.*

company_mast

com_id	**com_name**
11	Samsung
12	iBall
13	Epsion
14	Zebronics
15	Asus
16	Frontech

item_mast

pro_id	**pro_name**	**pro_price**	**pro_com**
101	Mother Board	3200.00	15
102	Key Board	450.00	16
103	Zip Drive	250.00	14
104	Speaker	550.00	16
105	Monitor	5000.00	11
106	DVD	900.00	12
107	CD	800.00	12
108	Printer	2600.00	13
109	Refill Cartridge	350.00	13
110	Mouse	250.00	12

Show the output for the following queries.

(a)
```
SELECT AVG(pro_price), company_mast.com_name FROM item_mast INNER
JOIN company_mast
ON item_mast.pro_com= company_mast.com_id
GROUP BY company_mast.com_name
HAVING AVG(pro_price) >= 350;
```

ble which also
name of the ~~company~~
able we returned the ~~company~~
which employs more than 2
employees.

which also
n the

```
ELECT A.pro_name, A.pro_price, F.com_name FROM item_mast A
NNER JOIN company_mast F
DN A.pro_com = F.com_id AND A.pro_price =
 SELECT MAX(A.pro_price) FROM item_mast A WHERE A.pro_com = F.com_id);
```

(3+3)

. Consider three tables *customer, salesman and orders* with customer_id, nan_id and ord_no as their Primary Keys respectively. salesman_id of *customer* is a gn Key referencing the salesman_id of *salesman*. customer_id of *orders* is a Foreign Key eferencing the customer_id of *customer*. salesman_id of *orders* is a Foreign Key rencing the salesman_id of *salesman*.

stomer

ustomer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3008	Julian Green	London	300	5002
3005	Graham Zusi	California	200	5002
3009	Geoff Cameron	Berlin	100	5003
3004	Fabian Johnson	Paris	300	5006
3001	Brad Guzan	London		5005
3003	Jozy Altidor	Moscow	200	5007

salesman

salesman_id	name	city	commission
5006	Mc Lyon	Paris	0.14
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5003	Lauson Hen	San Jose	0.12
5005	Pit Alex	London	0.11
5007	Paul Adam	Rome	0.13

orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2022-10-05	3005	5002
70011	75.29	2022-08-17	3003	5007
70009	270.65	2022-09-10	3001	5005
70002	65.26	2022-10-05	3002	5001
70005	2400.6	2022-07-27	3007	5001
70004	110.5	2022-08-17	3009	5003
70007	948.5	2022-09-10	3005	5002
70013	3045.6	2022-04-25	3002	5001
70008	5760	2022-09-10	3002	5001
70010	1983.43	2022-10-10	3004	5006
70003	2480.4	2022-10-10	3009	5003
70012	250.45	2022-06-27	3008	5002

(a) Show the output for:

(i) SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.city, b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission>.12 AND a.city<>b.city;

(ii) SELECT a.cust_name, a.city, a.grade, b.name AS "Salesman", c.ord_no, c.ord_date, c.purch_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id LEFT OUTER JOIN orders c ON c.customer_id=a.customer_id WHERE c.purch_amt>=2000 AND a.grade IS NOT NULL;

(b) How many tuples will have city as '*London*' on executing the following query? SELECT a.cust_name, a.city, b.ord_no, b.ord_date, b.purch_amt AS "Order Amount" FROM customer a FULL OUTER JOIN orders b ON a.customer_id=b.customer_id WHERE a.grade IS NOT NULL;

(3+3+3)

Ques 5. Given a relation BOOK(ISBN, Title, Publisher, Address) and Functional Dependency set (ISBN → Title, ISBN → Publisher, Publisher → Address). Determine the normal form of the given relation.

(3)

Ques 6. Should all data models be normalized to 3NF? If so, why? If not, give an example where 3NF would cause issues. (3)

Ques 7. Refer to the following tables:

StudentDetails

StudId	Name	EnrollmentNo	DateOfJoining
11	Nick Panchal	1234567	01/02/2019
21	Yash Panchal	2468101	15/03/2017
31	Gyan Rathod	3689245	27/05/2018

StudentStipend

StudId	Project	Stipend
11	P1	80000
21	P2	10000
31	P1	120000

Write an SQL query to:

(a) Fetch student names and stipend records. Return student details even if the stipend record is not present for the student.

(b) Fetch all student records from StudentDetails table who have a stipend record in StudentStipend table.

(c) Retrieve all the Students who also have enrollment No from StudentDetails table.

(d) Fetch count of students project-wise sorted by project's count in descending order.

(e) Find the nth highest stipend from the table.

(3*5=15)