COMPSCI 351	The University of Auckland + Southwest University

Fundamentals of Database Systems - Assignment 3

2024

Note: Collaboration on assignments is encouraged, but you must write up your work individually and in your own words.

1. **(Entity-Relationship Modelling)**

   Consider the relational database schema {`CUSTOMER`, `ORDER`, `DRIVER`} with the relation schemata:

   - `CUSTOMER`={*name, dob, email*} with key {name, dob}

   - `DRIVER`={*date, driver_name, vehicle*} with key {*date*}

   - `ORDERS`={*order_id, name, dob, address, payment_method, date*} with key {order_id, name, dob} and with foreign keys

     [name, dob] $\subseteq$ `CUSTOMER`[name, dob]

     [date] $\subseteq$ `DRIVER`[date]

   (a) Specify the ER schema that corresponds to the database schema above.

   [4 marks]

   (b) Specify the ER diagram that corresponds to the database schema above.

   [2 marks]

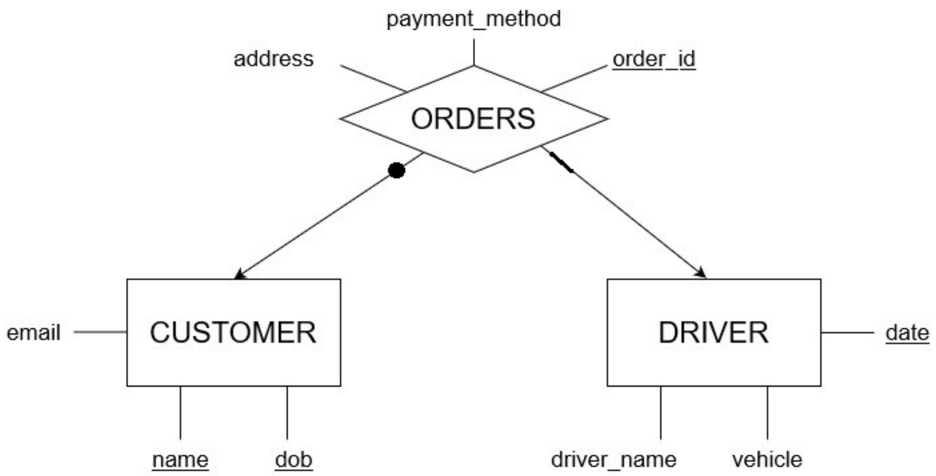
   (c) Define an instance over the ER schema that contains at least three objects for each object type.
   i) Use foreign key semantics to specify your instances.
   ii) In addition, use identifier semantics to write down the same instances.

   [3 marks]

SOLUTION:

(a) - `CUSTOMER`=({*name, dob, email*}, {*name, dob*})
- `DRIVER`=({*date, driver_name, vehicle*}, {*date*})
- `ORDERS`=({`CUSTOMER`, `DRIVER`}, {*order_id, address, payment_method*}, {`CUSTOMER`, *order_id*})

(b) See the Entity Relationship Diagram below:



(c) i)

CUSTOMER		
name	dob	email
Alan Turing	1912-06-23	a.turing@np-computations.com
Peter Chen	1947-01-03	p.chen@erm.com
Edgar Codd	1923-08-19 3	e.codd@bcnf-computations.com

DRIVER		
date	driver_name	vehicle
2024-01-01	Alice	scooter
2024-02-01	Bob	scooter
2024-03-01	Alice	van

(In the following table 'pm' stands for 'payment_method' and has been abbreviated for formatting reasons)

ORDERS					
order_id	CUSTOMER		address	pm	DRIVER
1	Peter Chen	1947-01-03	Entity Street 5	cash	2024-01-01
2	Edgar Codd	1923-08-19	Relational Avenue 1	cash	2024-02-01
3	Edgar Codd	1923-08-19	Relational Avenue 1	online	2024-03-01

ii)

CUSTOMER			
ID	name	dob	email
c1	Alan Turing	1912-06-23	a.turing@np-computations.com
c2	Peter Chen	1947-01-03	p.chen@erm.com
c3	Edgar Codd	1923-08-19 3	e.codd@bcnf-computations.com

DRIVER			
ID	date	driver_name	vehicle
d1	2024-01-01	Alice	scooter
d2	2024-02-01	Bob	scooter
d3	2024-03-01	Alice	van

ORDERS					
ID	order_id	CUSTOMER	address	payment_mehtod	DRIVER
o1	1	c2	Entity Street 5	cash	d1
o2	2	c3	Relational Avenue 1	cash	d2
o3	3	c3	Relational Avenue 1	online	d3

2. **(Functional Dependencies)**

Consider the following relation over the relation schema `ADDRESS`:

number	street	zip	city
88	The Strand	1010	Auckland
38	Liverpool Street	1026	Auckland
38	Princes Street	1010	Auckland
42	Princes Street	6021	Wellington
38	Cuba Street	6021	Wellington

(a) Find a non-trivial functional dependency that holds on the relation, and perform a lossless decomposition of the relation using the functional dependency.

[2 marks]

(b) In the decomposition from a) we managed to eliminate redundancy. Give an example of the problems that redundancy in relational databases can cause.

[1 marks]

SOLUTION:

(a) A non-trivial FD is zip $\rightarrow$ city and we can decompose the relation losslessly into:

number	street	zip
88	The Strand	1010
38	Liverpool Street	1026
38	Princes Street	1010
42	Princes Street	6021
38	Cuba Street	6021

zip	city
1010	Auckland
1026	Auckland
6021	Wellington

(b) Redundancy can cause the following issues:

- Updates can be slower as multiple (redundant copies of) values have to be updated instead of just one

- Redundancy can cause anomalies within the data when for example updating only some of the redundant copies of the respective values

- Redundancy can cause anomalies within the data when for example inserting that records violate some FD

3. **(Database Normalization)**

Consider the relation schema FACULTY = $\{$*lecturer*, *department_head*, *department*$\}$ which keeps information on a university faculty along with the lecturers, the different departments and the head of department. In addition, we have the set of functional dependencies that hold on FACULTY which is $\Sigma = \{lecturer \to department,\ department\_head \to department\}$.

(a) Determine all keys for FACULTY with respect to Σ and explain why there cannot be any further keys.

[2 marks]

(b) Is the schema FACULTY in 3NF (with respect to Σ)? Please explain why or why not. If not, determine a faithful, lossless 3NF decomposition of FACULTY.

[3 marks]

(c) Is the schema FACULTY in BCNF (with respect to Σ)? Please explain why or why not. If not, determine a lossless BCNF decomposition of FACULTY and explain whether or not this is also faithful.

[3 marks]

SOLUTION:

(a) We use the closure algorithm systematically for different sets of attributes.

- $\{lecturer\}^+ = \{lecturer, department\} \neq \{lecturer, department\_head, department\}$ is not superkey

- $\{department\_head\}^+ = \{department\_head, department\} \neq \{lecturer, department\_head, department\}$ is not superkey

- $\{department\}^+ = \{department\} \neq \{lecturer, department\_head, department\}$ is not superkey

- $\{lecturer, department\_head\}^+ = \{lecturer, department\_head, department\}$ is superkey. As neither subset, i.e. $\{lecturer\}$ or $\{department\_head\}$ are superkeys already this means that $\{lecturer, department\_head\}$ is key.

- $\{lecturer, department\}^+ = \{lecturer, department\} \neq \{lecturer, department\_head, department\}$ is not superkey

- $\{department\_head, department\}^+ = \{department\_head, department\} \neq \{lecturer, department\_head, department\}$ is not superkey

- $\{lecturer, department\_head, department\}$ is trivially a superkey (in a relation schema), however not key as the subset $\{lecturer, department\_head\}$ is already key.

(b) The schema FACULTY is not in 3NF.

For example if we consider the FD *lecturer* $\to$ *department* we note that the LHS *lecturer* does not constitute a superkey as shown in a) and neither is the RHS *department* a prime attribute as it is not contained in any key, i.e. in the only key $\{lecturer, department\_head\}$ as shown in a).

Using the synthesis algorithm (where Σ is already canonical cover) we get

- $R_1 = \{lecturer, department\}$ with FD set $\Sigma_1 = \{lecturer \to department\}$
- $R_2 = \{department\_head, department\}$ with FD set $\Sigma_2 = \{department\_head \to department\}$
- $R_3 = \{department\_head, lecturer\}$ with FD set $\Sigma_3 = \emptyset$

(We note that R_3 is required to be able to verify that $\{department\_head, lecturer\}$ still is key, i.e. to keep the decomposition lossless)

(c) As BCNF is a special case of 3NF (3NF is a generalisation of BCNF) and `FACULTY` is not in 3NF as shown in b), it cannot be in BCNF.

The BCNF decomposition algorithm gives us:

- $R_1 = \{lecturer,\ department\}$ with FD set $\Sigma_1 = \{lecturer \rightarrow department\}$
- $R_2 = \{department\_head,\ lecturer\}$ with FD set $\Sigma_2 = \emptyset$

Alternatively, starting with the other FD for the decomposition we could get:

- $R_1' = \{department\_head,\ department\}$ with FD set $\Sigma_1' = \{department\_head \rightarrow department\}$
- $R_2' = \{department\_head,\ lecturer\}$ with FD set $\Sigma_2' = \emptyset$

In either case the decomposition is not faithful as for the decomposition $\{(R_1, \Sigma_1), (R_2, \Sigma_2)\}$ we lost the FD *department_head* $\rightarrow$ *department* and would need to join relations to verify if this FD is satisfied and for the alternative solution $\{(R_1', \Sigma_1'), (R_2', \Sigma_2')\}$ we would lose the FD *lecturer* $\rightarrow$ *department* which we would not be able to verify without joining the relations.

Possible Marks: 20