COMP2400/6240 - Relational Databases

Due date: 23:59, 11 October, 2022

Instructions:

- **This assignment must be done individually (no group work)**.
- This assignment will count for 15% of the final grade. Marks are assigned for the process of finding a solution, not only for the result. Hence, include all essential ideas and steps that are necessary to derive a solution.
- You must submit a single PDF file named as "u1234567.pdf" (replace u1234567 with your UID). Make sure you only upload a PDF file, not a Word or text file.
- You should try your best to type the solutions. The scanned images of handwritten texts and equations can be unreadable for marking. As for the EER diagram, you are highly recommended to export a JPEG file from TerraER and include it in the PDF file.
- Late submission is not granted under any circumstance. You will be marked on whatever you have submitted at the time of the deadline. Please take careful note of deadlines and adhere to them. Of course, if you find yourself in a situation beyond your control that you believe significantly affects an assessment, you should send an Email to Yu Lin `<yu.lin@anu.edu.au>` with the title "Special Consideration for Assignment 2 (Database Theory)" along with the supporting documents.
- Plagiarism will attract academic penalties in accordance with the ANU guidelines. A student in this course is expected to be able to explain and defend any submitted assessment item. The course convener can conduct or initiate an additional interview about any submitted assessment item for any student. If there is a significant discrepancy between the two forms of assessment, it will be automatically treated as a case of suspected academic misconduct.

Question 1 **3 Marks**

Band-Aid is a booking agency that represents local Canberra bands by booking their shows nationally. The employees of Band-Aid are classified into two non-overlapping categories, agents and administrators. An agent represents one or more bands and can be contacted via their personal office phone or work email. Each administrator belongs to a department which offers administrative support to all agents of the agency. Note that the internal relationship between agents and administrators isn't something Band-Aid cares to track. Administrative departments include accounts, contracts, and promotions. Each Band-Aid employee can be uniquely identified by their employee number.

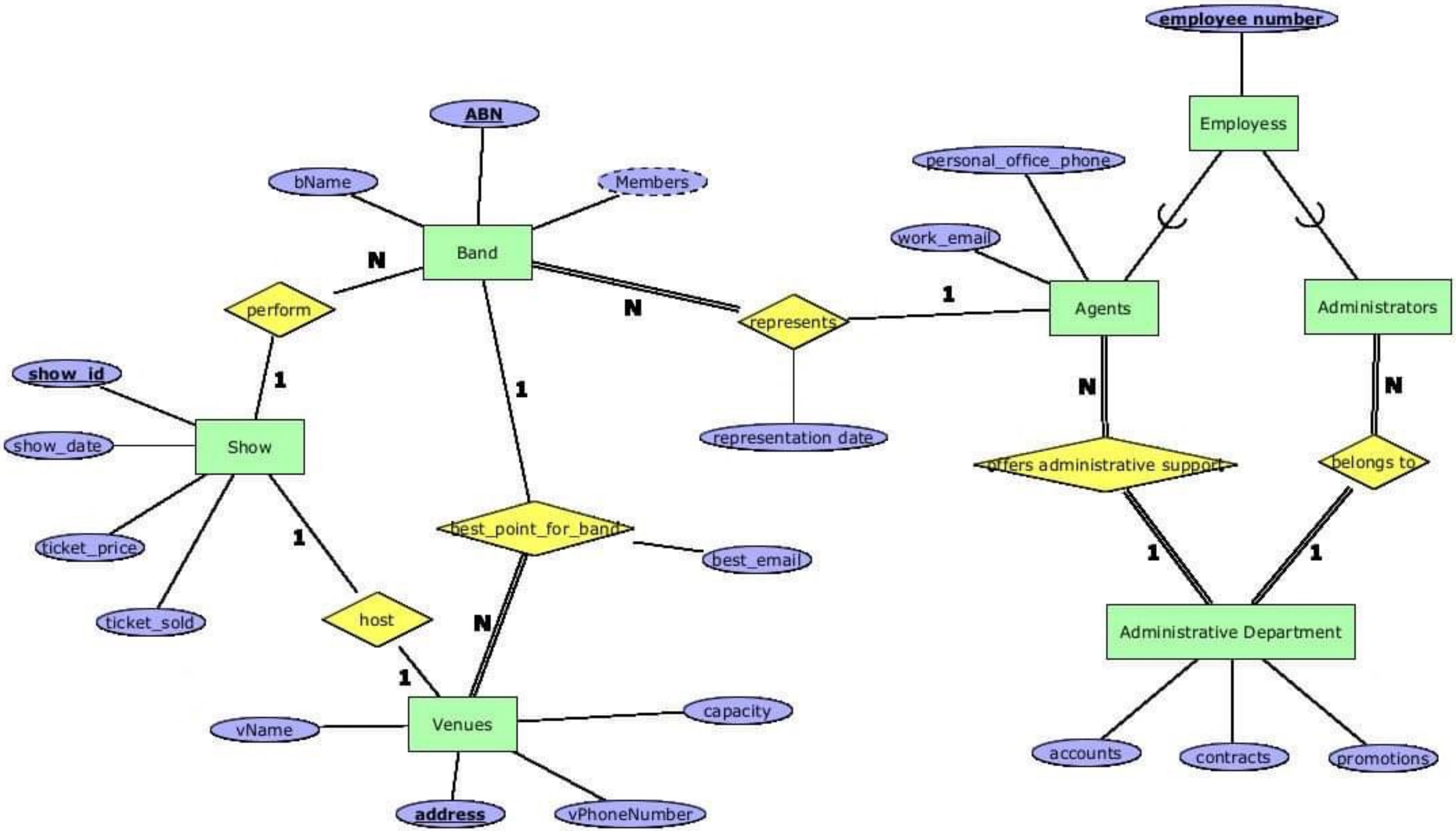
Each band has a name, can be uniquely identified by their Australian Business Number (ABN), and may have a number of band members. Band-Aid also stores the name, phone, and email of the best point of contact for the band. Each band is represented by a particular agent and Band-Aid likes to record the date that representation began. If a band has just joined the agency (or they aren't very popular!) then the agency may never have booked a show for the band.

Band-Aid maintains a list of venues. Each record includes the venue's name, a distinct address, phone number, and the venue capacity. Band-Aid likes to keep a comprehensive list of venues, including new venues or venues where they're yet to book a show.

The primary purpose of Band-Aid is to book shows. Each show is hosted by a particular venue. One or more of the bands Band-Aid represents would perform at the show. When Band-Aid books a show, they record the date and ticket price, and once the show is finalised, the number of tickets sold. Each show is assigned a unique show ID.

Your task is to design an Enhanced Entity Relationship (EER) diagram that captures the above requirements. It should include entities, relationships, attributes, and constraints where appropriate. You may make more assumptions, if necessary, but any assumptions should be noted. If there are any requirements that cannot be captured in an EER-diagram, then they should also be identified.

Q.1



employee number
Employess
ABN
bName
Members
Band
personal_office_phone
work_email
N
perform
N
represents
1
Agents
Administrators
show_id
1
1
N
N
show_date
Show
representation date
offers administrative support
belongs to
best_point_for_band
ticket_price
1
best_email
1
1
ticket_sold
host
N
1
Administrative Department
vName
Venues
capacity
accounts
contracts
promotions
address
vPhoneNumber

Question 2 **5 Marks**

Consider the relation schema R = $\{A, B, C, D, E, F, G\}$ and the following set Σ of FDs:

- $C \to BG$
- $AE \to F$
- $CF \to EA$
- $A \to DF$

2.1 What are the candidate keys of R? Justify your answer (i.e., include the main steps used for finding the candidate keys). (1 Mark)

2.2 Find a minimal cover of Σ and include the main steps used for finding a minimal cover. (2 Marks)

2.3 Demonstrate why R, given Σ, does not satisfy 3NF, and then identify a 3NF decomposition for R. You need to include the main steps used for identifying the 3NF decomposition. (2 Marks)

Question 3 **3 Marks**

Consider the relation schema BOOKING={Airline, Date, Destination, FlightNo, PassengerName, Origin, PassportNo, Seat, Terminal} and the following set Σ of FDs:

- Airline $\to$ Terminal
- FlightNo $\to$ Airline, Origin, Destination
- PassportNo $\to$ PassengerName
- FlightNo, PassportNo, Date $\to$ Seat
- FlightNo, Date, Seat $\to$ PassportNo

Which, for brevity, we can alias the attributes and restate as BOOKING={A, D, E, F, N, O, P, S, T} and Σ:

- A $\to$ T
- F $\to$ AOE
- P $\to$ N
- FPD $\to$ S
- FDS $\to$ P

3.1 Is the above relation schema BOOKING in BCNF? If not, identify a BCNF decomposition for BOOKING and check if your BCNF decomposition is dependency-preserving. You need to include the main steps used for identifying the BCNF decomposition. (2 Marks)

3.2 Consider another set Σ_1 of FDs on BOOKINGS:

- Airline $\to$ Terminal
- FlightNo $\to$ Airline, Origin, Destination
- PassportNo $\to$ PassengerName
- FlightNo, PassportNo, Date $\to$ Terminal, Seat
- FlightNo, Date, Seat $\to$ PassportNo, PassengerName

Which can be similarly aliased as:

- A $\to$ T
- F $\to$ AOE
- P $\to$ N

Question 2 **5 Marks**

Consider the relation schema R = $\{A, B, C, D, E, F, G\}$ and the following set Σ of FDs:

- $C \to BG$
- $AE \to F$
- $CF \to EA$
- $A \to DF$

2.1 What are the candidate keys of R? Justify your answer (i.e., include the main steps used for finding the candidate keys). (1 Mark)

2.2 Find a minimal cover of Σ and include the main steps used for finding a minimal cover. (2 Marks)

2.3 Demonstrate why R, given Σ, does not satisfy 3NF, and then identify a 3NF decomposition for R. You need to include the main steps used for identifying the 3NF decomposition. (2 Marks)

2.1. $R = \{A, B, C, D, E, F, G\}$

$\Sigma: \{C \to BG, AE \to F, CF \to EA, A \to DF\}$

Solution: Check $(X)^+$ for every subset of R

- C never appears in the dependent of any FD, C must be the part of the key (From trick)

- B, D, G never appears in the determinant so B must not be part of the key

So consider

1n $(C)^+ = CBG$;

2n $(CA)^+ = ABCDEFG$; $(CF)^+ = ABCDEFG$

3n $(CAE)^+ = ABCDEFG$; $(CAF)^+ = ABCDEFG$; $(CEF)^+ = ABCDEFG$;

4n $(CAEF)^+ = ABCDEFG$

Candidate keys of R is $\{C, A\}$, $\{C, F\}$ ~~X~~

2.2 $\Sigma : \{C \to BG, AE \to F, CF \to EA, A \to DF\}$

1. start from $\Sigma_m : \{C \to BG, AE \to F, CF \to EA, A \to DF\}$

2. check RHS with one attribute

$\Sigma_m : \{C \to B, C \to G, AE \to F, CF \to E, CF \to A, A \to D, A \to F\}$

3. check LHS for minimized

For	$AE \to F$	to	$A \to F$	can't
	$AE \to F$	to	$E \to F$	can't
	$CF \to E$	to	$C \to E$	can't
	$CF \to E$	to	$F \to E$	can't
	$CF \to A$	to	$C \to A$	can't
	$CF \to A$	to	$F \to A$	can't

∴ can't replaced

4. $\Sigma_m = \{C \to BG, AE \to F, CF \to EA, A \to DF\}$

There fore it minimal cover itself. //

2.3. Assumption: FD non trivial $X \to A$ which
X is a superkey, A is a prime
attribute

From 2.1 We got (CA), (CF)
is a key and all attributes

Finding prime attributes is an
attributes occurring in key

so prime attributes is C, A, F
non-prime attributes is B, D, G, E

check all FD

: $C \to BG$ → C is not superkey
BG is not prime attribute

$AE \to F$ → AE is not superkey
F is prime.

$CF \to EA$ → CF is a superkey
EA is not prime

$A \to DF$ → A is not superkey
DF is not prime.

∴ So it obviously see a lot

of 3NF violation ~~A~~

so we will begin decomposition.

Solution:

- Finding minimal covers
  ↳ from 2.2 we will see it is already minimum cover it self
- so group the FD in minimal cover

$$\Sigma_m = \{C \to BG, AE \to F, CF \to EA, A \to DF\}$$

↓ Group

$$S = \{\underset{R_1}{CBG}, \underset{R_2}{AEF}, \underset{R_3}{CFEA}, \underset{R_4}{ADF}\}$$

R_3 omit R_2

$$S = \{\underset{R_1}{CBG}, \underset{R_3}{CAEF}, \underset{R_4}{ADF}\}$$

since R_3 is superkey we don't need to add key

Thus $S = \{R_1, R_3, R_4\}$

so we got decomposed

$R_1 = \{CBG\}$ with $\Sigma_1 \{C\} \to \{BG\}$

$R_3 = \{CAEF\}$ with $\Sigma_3 = \{CF\} \to \{EA\}$

$R_4 = \{ADF\}$ with $\Sigma_4 = \{A\} \to \{DF\}$

so $\{\Sigma_1 \cup \Sigma_3 \cup \Sigma_4\}$ is equivalent to Σ

The 3NF-decomposition is

$\{CBG, CFEA, ADF\}$ X

Question 3 **3 Marks**

Consider the relation schema BOOKING={Airline, Date, Destination, FlightNo, PassengerName, Origin, PassportNo, Seat, Terminal} and the following set Σ of FDs:

- Airline $\rightarrow$ Terminal
- FlightNo $\rightarrow$ Airline, Origin, Destination
- PassportNo $\rightarrow$ PassengerName
- FlightNo, PassportNo, Date $\rightarrow$ Seat
- FlightNo, Date, Seat $\rightarrow$ PassportNo

Which, for brevity, we can alias the attributes and restate as BOOKING={A, D, E, F, N, O, P, S, T} and Σ:

- A $\rightarrow$ T
- F $\rightarrow$ AOE
- P $\rightarrow$ N
- FPD $\rightarrow$ S
- FDS $\rightarrow$ P

3.1 Is the above relation schema BOOKING in BCNF? If not, identify a BCNF decomposition for BOOKING and check if your BCNF decomposition is dependency-preserving. You need to include the main steps used for identifying the BCNF decomposition. (2 Marks)

3.2 Consider another set Σ_1 of FDs on BOOKINGS:

- Airline $\rightarrow$ Terminal
- FlightNo $\rightarrow$ Airline, Origin, Destination
- PassportNo $\rightarrow$ PassengerName
- FlightNo, PassportNo, Date $\rightarrow$ Terminal, Seat
- FlightNo, Date, Seat $\rightarrow$ PassportNo, PassengerName

Which can be similarly aliased as:

- A $\rightarrow$ T
- F $\rightarrow$ AOE
- P $\rightarrow$ N
- FPD $\rightarrow$ TS
- FDS $\rightarrow$ PN

Are Σ and Σ_1 equivalent? Justify your answer. (1 Mark)

Assumption: Finding keys.

Booking = {A, D, E, F, N, O, P, S, T}, FDs: A→T, F→AOE, P→N, FPD→S, FDS→P

D, F never appear in RHS so DF must be candidate keys

2n $(DF)^+ =$ AOEDFT;

3n $(DFP)^+ =$ AOEDFTPSN;

$(DFS)^+ =$ AOEDFTPSN;

so keys are {(DFP), (DFS)}

so FD on FPD, FPS is superkey

but A, F, P is not a superkey

so it's not in BCNF #

we need to decompose

so we decompose non BCNF relation

$\{A \to T\}, \{F \to AOE\}, \{P \to N\}$

$\downarrow \quad - \{A \to T\}$

$\{AT, ADEFNOPS\}$

$\downarrow \quad - \{F \to AOE\}$

$\{AT, FAOE, DFNPS\}$

$\downarrow \quad - \{P \to N\}$

$\{AT, FAOE, PN, FDPS\}$

Now the BCNF final relation is

(AT), (AFOE), (PN), (FDPS)} #

super key

3.2. Compare to 3.1 key is still

$\{(DFP)^+, (DFS)^+\}$ - super key

A, F, P is not super key

so we need to decompose.

$\{A \to T\}, \{F \to AOE\}, \{P \to N\}$

$\downarrow$

so

BCNF final relation is

$(AT), (AFOE), (PN), (FDPS)\}$

so Σ_1 is Σ ✗

- FPD $\rightarrow$ TS
- FDS $\rightarrow$ PN

Are Σ and Σ_1 equivalent? Justify your answer. (1 Mark)

Question 4 **4 Marks**

The following table contains the relational algebra operators covered in our course. You should only use these operators to answer the following questions.

$\sigma_{\varphi}R$	Selection by condition φ
$\pi_{A_1,\dots,A_n}R$	Projection onto the set of attributes $\{A_1 \dots, A_n\}$
$\rho_{R'(A_1,\dots,A_n)}R$	Renaming the relation name to $R^{'}$ and attribute names to $A_1, \dots, A_n$
$\rho_{R'}R$	Renaming the relation name to $R^{'}$
$\rho_{(A_1,\dots,A_n)}R$	Renaming the attribute names to $A_1, \dots, A_n$
$R_1 \cup R_2$	Union of two relations R_1 and R_2
$R_1 \cap R_2$	Intersection of two relations R_1 and R_2
$R_1 - R_2$	Difference of two relations R_1 and R_2
$R_1 \times R_2$	Cartesian product of two relations R_1 and R_2
$R_1 \bowtie_{\varphi} R_2$	Join of two relations R_1 and R_2 with the join condition φ
$R_1 \bowtie R_2$	Natural join of two relations R_1 and R_2
$\varphi_1 \bigwedge \varphi_2$	condition φ_1 AND condition φ_2
$\varphi_1 \bigvee \varphi_2$	condition φ_1 OR condition φ_2

Consider a relational database schema $\mathcal{S}$ with the following relation schemas:

- Movie(title, production_year, genre) with the primary key {title, production_year}
- Person(id, first_name, last_name, year_born) with the primary key {id}
- Director(id, title, production_year) with the primary key {title, production_year} and with the foreign keys [title, production_year] $\subseteq$ Movie[title, production_year] and [id] $\subseteq$ Person[id]
- Writer(id, title, production_year, credits) with primary key {id, title, production_year} and with the foreign keys [title, production_year] $\subseteq$ Movie[title, production_year] and [id] $\subseteq$ Person[id]

4.1 Answer the following questions using relational algebra queries only using the operators in the above table. (2 Marks)

[a] List the ids, first and last names of directors who have never written any movie(s). (1 Mark)

[b] A person has worked on a movie if this person is a director, a writer, or both a director and writer, of this movie. List the ids of persons who have worked on at least two distinct movies. (1 Mark)

4.2 Optimise the following relational algebra query. Your marks will depend on how well you optimise the query in your solution. Additionally, draw the query trees of the query before and after your optimisation. (2 Marks)

$$\pi_{\text{Director}.id}\sigma_{(\text{Director}.id=\text{Person}.id) \bigwedge (year\_born > production\_year-40)}(\text{Director} \times \text{Person})$$
$$-\pi_{id}\sigma_{(year\_born<1970) \bigvee (production\_year>2010)}(\text{Writer} \bowtie \text{Person})$$

+++++

foreign keys [title, production_year] ⊆ MOVIE[title, production_year] and [id] ⊆ PERSON[id]

4.1 Answer the following questions using relational algebra queries only using the operators in the above table. (2 Marks)

[a] List the ids, first and last names of directors who have never written any movie(s). (1 Mark)

$R_1 := \pi_{id,\ title}\ (\text{Director})$

$R_2 := \pi_{id,\ title}\ (\text{Writer})$

$R_3 :\ \pi_{id,title}\ (R_1 - R_2)$.

$\pi_{id.\text{Person},\ fname.\text{Person},\ lname.\text{Person}}\ (\text{Person} \bowtie_{\text{Person}.id = R_3.id} R_3)$

[b] A person has worked on a movie if this person is a director, a writer, or both a director and writer, of this movie. List the ids of persons who have worked on at least two distinct movies. (1 Mark)

$R_1 := \pi_{id,title,production\_year}\ (\text{Director} \bowtie_{\text{Director}.id = \text{Writer}.id} \text{Writer})$

$R_2 := \pi_{id,\ title,\ production\_year}\ (\text{Director})$

$R_3 := \pi_{id,\ title,\ production.year}\ (\text{Writer})$

$R_4 = \ R_2 - R_1$ ← Director only

$R_5 = \ R_3 - R_1$ ← Writer only

$R_6 = \ R_2 - (R_4 \cup R_5)$ ← worked both

Result := $\pi_{id}\ (\text{Person}) - \pi_{id}\ (R_6 \cup R_4 \cup R_5)$

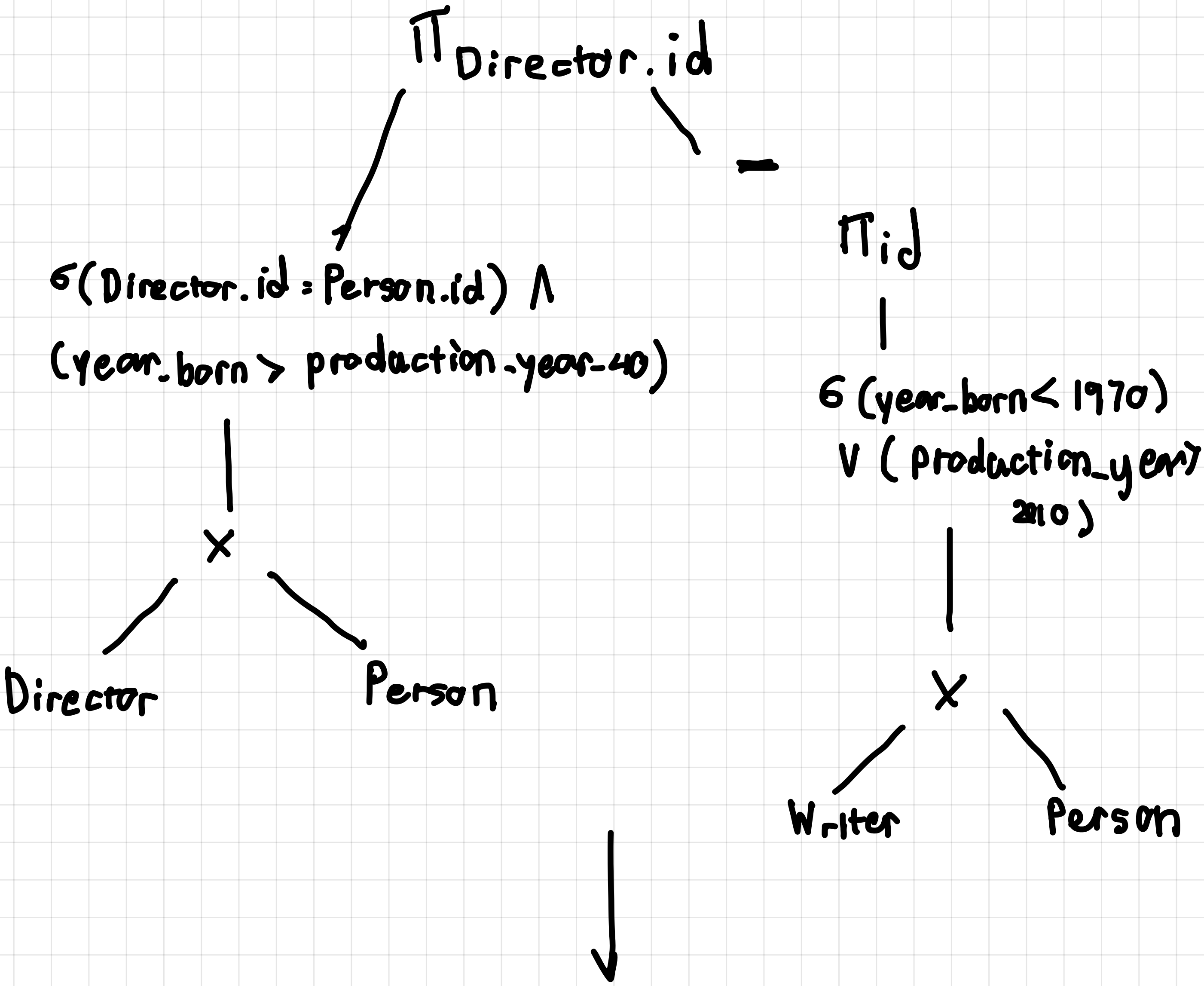
↑

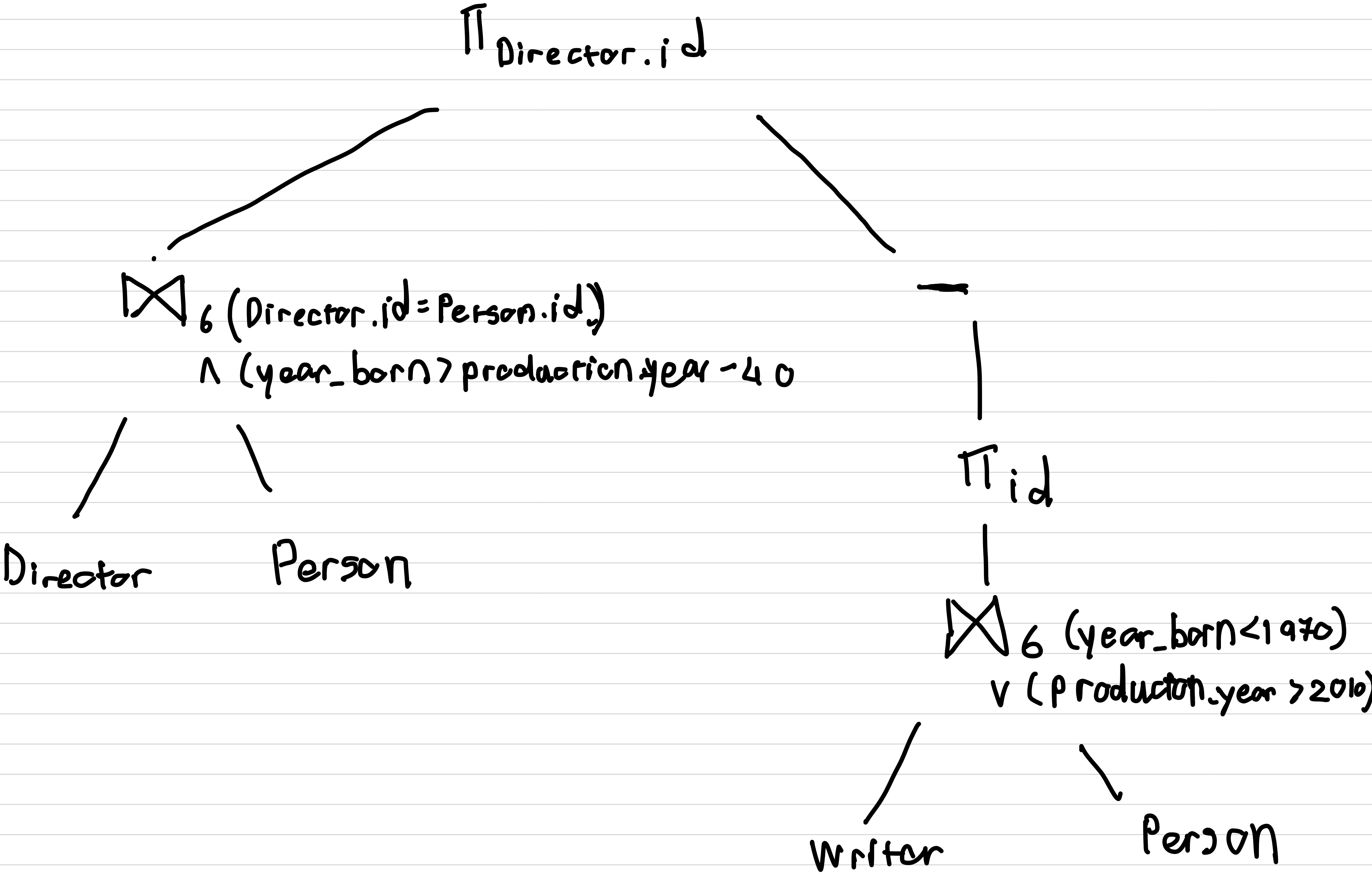
set of who worked at most one

4.2 Optimise the following relational algebra query. Your marks will depend on how well you optimise the query in your solution. Additionally, draw the query trees of the query before and after your optimisation. (2 Marks)

$$\pi_{\text{DIRECTOR}.id}\sigma_{(\text{DIRECTOR}.id=\text{PERSON}.id)\wedge(year\_born>production\_year-40)}(\text{DIRECTOR}\times\text{PERSON})$$
$$-\pi_{id}\sigma_{(year\_born<1970)\vee(production\_year>2010)}(\text{WRITER}\bowtie\text{PERSON})$$

+++++





Π Director.id
⋈ σ (Director.id=Person.id)
∧ (year_born > production.year - 40
Director
Person
−
Π id
⋈ σ (year_born<1970)
∨ (production.year > 2010)
Writer
Person