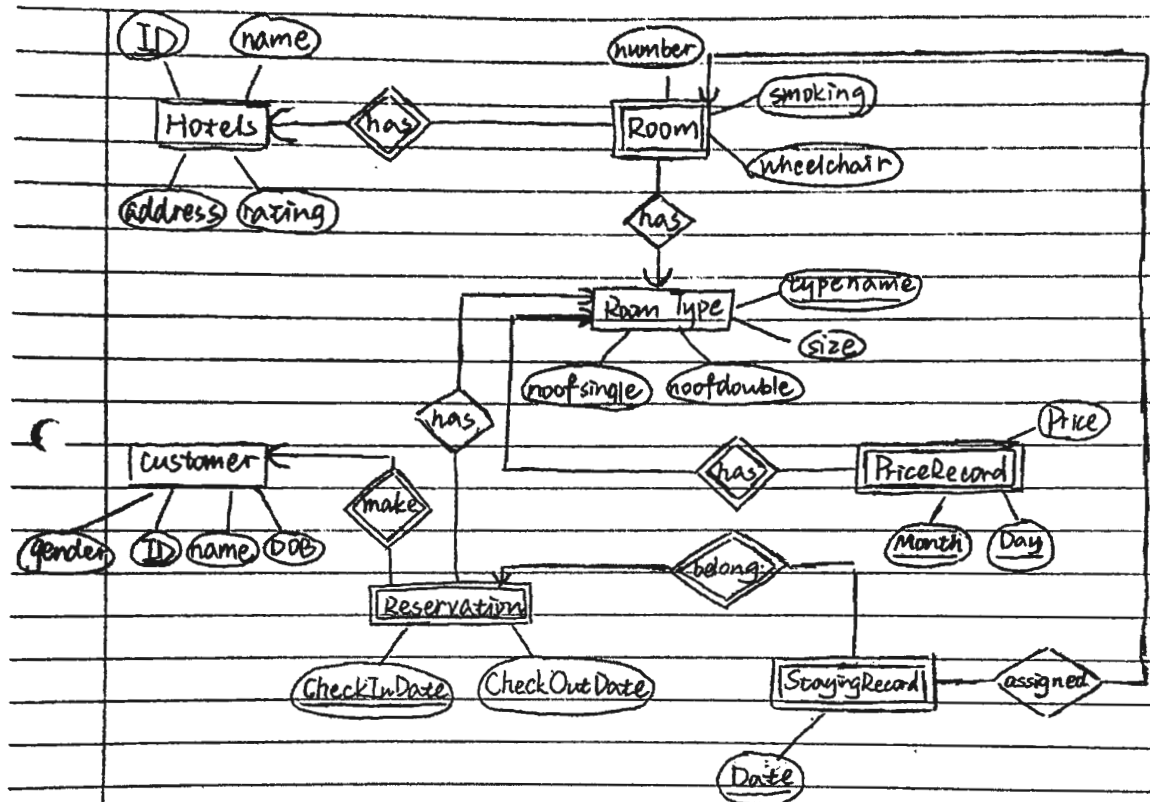


solver: Huang Xuhui

Email Address: C150177@e.ntu.edu.sg

1. (a) (i)



(ii)

Hotels (ID, name, address, rating)Room (HotelID, number, roomtype, smoking, wheelchair)RoomType (typename, size, noOfSingle, noOfDouble)PriceRecord (typename, month, day, price)Customer (ID, gender, name, DOB)Reservation (customerID, checkInDate, typename, checkOutDate)StayingRecord (date, customerID, checkInDate, hotelID, roomNumber)

(b) (i)

R1 := $\sigma_{\text{source} = \text{"Seoul" and destination} = \text{"Busan"}}$ TRAINR2 := Π_{TID} R1R3 := R2 \bowtie TICKETRESULT := $\gamma_{\text{count}(\text{TID}) \rightarrow \text{Passengers}}$ R3

(ii)

R1 := $\sigma_{\text{source} = \text{"Seoul" and destination} = \text{"Busan"}}$ TRAINR2 := $\sigma_{\text{source} = \text{"Busan" and destination} = \text{"Seoul"}}$ TRAINR3 := R1 \bowtie R1.Date = R2.Date R2RESULT := $\Pi_{(TID, PID)} \text{TICKET} \div \Pi_{TID} R3$

(iii)

R1 := $\gamma_{TID, \text{count}(PID) \rightarrow \text{numOfBooked}}$ TICKETR2 := R1 \bowtie R1.numOfBooked = TRAIN.NumberOfSeats and R1.TID = TRAIN.TID TRAINRESULT := $\Pi_{TID} R2$

(iv)

R1 := $\gamma_{PID, \text{count}(TID) \rightarrow \text{travelTimes}}$ TICKETR2 := $\Pi_{PID} (\sigma_{\text{travelTimes} > 100} R1)$ R3 := TICKET \bowtie TICKET.PID = R2.PID R2R4 := $\gamma_{TID, \text{count}(PID) \rightarrow \text{numOfBooked}}$ R3

R5 := R4

R6 := R4 \bowtie R4.TID \neq R5.TID and R4.numOfBooked < R5.numOfBooked R5R7 := $\Pi_{TID} R4 - \Pi_{TID} R6$ RESULT := $\Pi_{\text{source, destination}} (R7 \bowtie \text{TRAIN})$

2. (a) FDs: $AF \rightarrow D$, $B \rightarrow C$, $BE \rightarrow D$, $BDA \rightarrow F$, $CE \rightarrow A$, $D \rightarrow E$
 B is not in right hand side of FDs, B must be contained in the key

 $\{B\}^+ = \{B, C\}$, $\{A, B\}^+ = \{A, B, C\}$, $\{B, C\}^+ = \{B, C\}$, $\{B, D\}^+ = \{B, C, D, E, A, F\}$, $\{B, E\}^+ = \{B, E, C, A, D, F\}$, $\{B, F\}^+ = \{B, F, C\}$, $\{A, B, C\}^+ = \{A, B, C\}$, $\{A, B, F\}^+ = \{A, B, F, D, E, C\}$, $\{B, C, F\}^+ = \{B, C, F\}$

Key: BD, BE, ABF

 $AF \rightarrow D$ violated BCNF. $\{A, F, D\}^+ = \{A, F, D, E\}$ R (A, B, C, D, E, F) \rightarrow R1 (A, F, D, E) and R2 (A, F, D, B, C) $B \rightarrow C$ violated BCNF. $\{B, C\}^+ = \{B, C\}$ R2 (A, F, D, B, C) \rightarrow R3 (B, C) and R4 (A, F, D, B) $D \rightarrow E$ violated BCNF. $\{D, E\}^+ = \{D, E\}$ R1(A, F, D, E) \rightarrow R5 (A, F, D) and R6 (D, F)

For R4 (A, F, D, B)

$\{B, C\},$
 $\{D\}^+ = \{D, E\},$
 $\{F\}^+ = \{F\},$
 $\{A, B\}^+ = \{A, B, C\},$
 $\{A, D\}^+ = \{A, D, E\},$
 $\{A, F\}^+ = \{A, F, D\}$
 $R4 (A, F, D, B) \rightarrow R7 (A, F, B) \text{ AND } R8 (A, F, D)$

FDs that is not preserved : $C, E \rightarrow A$

(b) FDs: $AF \rightarrow D, B \rightarrow C, BE \rightarrow D, BDA \rightarrow F, CE \rightarrow A, D \rightarrow E$
 Try to remove $AF \rightarrow D, \{A, F\}^+ = \{A, F\}, AF \rightarrow D$ cannot be remove.
 Try to remove $B \rightarrow C, \{B\}^+ = \{B\}, B \rightarrow C$ cannot be remove.
 Try to remove $BE \rightarrow D, \{B, E\}^+ = \{B, E, C, A\}, BE \rightarrow D$ cannot be remove.
 Try to remove $BDA \rightarrow F, \{B, D, A\}^+ = \{B, D, A, C, E\}, BDA \rightarrow F$ cannot be remove.
 Try to remove $CE \rightarrow A, \{C, E\}^+ = \{C, E\}, CE \rightarrow A$ cannot be remove.
 Try to remove $D \rightarrow E, \{D\}^+ = \{D\}, D \rightarrow E$ cannot be remove.

Next step try to remove the redundant attribute of left hand side.
 Only A can be removed from $BDA \rightarrow F$

Now the FDs: $AF \rightarrow D, B \rightarrow C, BE \rightarrow D, BD \rightarrow F, CE \rightarrow A, D \rightarrow E$
 Finally, R (A, B, C, E, F) is normalized into R1 (A, F, D), R2 (B, C), R3 (B, E, D), R4 (B, D, F)
 R5 (C, E, A)

3. (a) (i)

select categoryname from Category as ct1 where not exists (select
 categoryname from Category as ct2 where ct1.categoryname =
 ct2.belongsto)

(ii)

select Borrowed.ReaderNr from (select Loan.ReaderNr, Loan.ISBN from
 Loan, Book where Loan.ISBN = Book.ISBN and Book.author = 'Jiawei
 Han' group by Loan.ReaderNr, Loan.ISBN) as Borrowed group by
 Borrowed.ReaderNr having count(Borrowed.ReaderNr) = (select
 COUNT(ISBN) from Book where author = 'Jiawei Han')

Note: question (ii) may has a simpler solution, the above one is the
 one I write in the exam

(iii)

select DISTINCT ISBN from Copy where ISBN in (select Rest.ISBN
 from (select Copy.ISBN, Copy.copynumber from Copy except select
 ISBN, copy from Loan) AS Rest)

CZ2007 - INTRODUCTION TO DATABASE

(b)

CHECK: Create ASSERTION noMoreThan20 AS Check(NOT exists((select count(ReaderNr) from Loan group by ReaderNr) >20));

TRIGGER:

Create TRIGGER noMoreThan20

AFTER INSERT ON Loan

REFERENCING NEW ROW AS new FOR

EACH ROW

WHEN (select count(ReaderNr) from Loan group by ReaderNr) >20

ROLLBACK

(c)Update Loan

SET ReturnDate = DATEADD(day, 30, ReturnDate)

Where ReturnDate < '15.03.2013'

4. (a) Left outer join

$$R \bowtie_L S =$$

$$(R \bowtie S) \cup ((R - \pi_{r_1, r_2, \dots, r_n}(R \bowtie S)) \times \{(\omega, \dots, \omega)\})$$

r_1, r_2, \dots, r_n are the attributes of R , the ω is the null value (those that are not attributes of R)

Right outer join

$$R \bowtie_R S =$$

$$(R \bowtie S) \cup (\{(\omega, \dots, \omega)\} \times (S - \pi_{s_1, s_2, \dots, s_n}(R \bowtie S)))$$

s_1, s_2, \dots, s_n are the attributes of S , the ω is the null value (those that are not attributes of S)

Full outer join

$$R \bowtie_{Full} S = (R \bowtie_L S) \cup (R \bowtie_R S)$$

More details can be found:

https://en.wikipedia.org/wiki/Relational_algebra#Left_outer_join_.28E2.9F.95.29

(b) Let's use A, B, C, D, E to present the Student_ID, Date_Enrolled, Course_ID, Room NR, Professor respectively. Now we have $R(A, B, C, D, E)$, $R_1(A, B, C)$ and $R_2(B, D, E)$. From description, we have $C \rightarrow E$, $C \rightarrow D$ and $A, C \rightarrow B$

(i) R

A	B	C	D	E
1	2	1	2	3
3	2	2	3	4

R_1

A	B	C
1	2	1
3	2	2

R_2

B	D	E
2	2	3
2	3	4

CZ2007 - INTRODUCTION TO DATABASE

R1 \bowtie R2

A	B	C	D	E
1	2	1	2	3
1	2	1	3	4
3	2	2	2	3
3	2	2	3	4

(ii) R1 (C, D, E), R2 (A, B, C)

C is super key of R1, so it's lossless decomposition

(c) (i)

1	1100
2	2200
3	3300
4	4400

(ii)

1	4400
2	4400
3	4400
4	4400

(d) (i) `/*[@choiceNum = "1"]/[@meritScore > 800]`(ii) `/*[@applicant = /*[@name = "Doreen"]/@appNum]/@code`

Note: Redo this paper a few months after my exam, if there are any errors please send me an email.

Thank you and all the best for your final exam