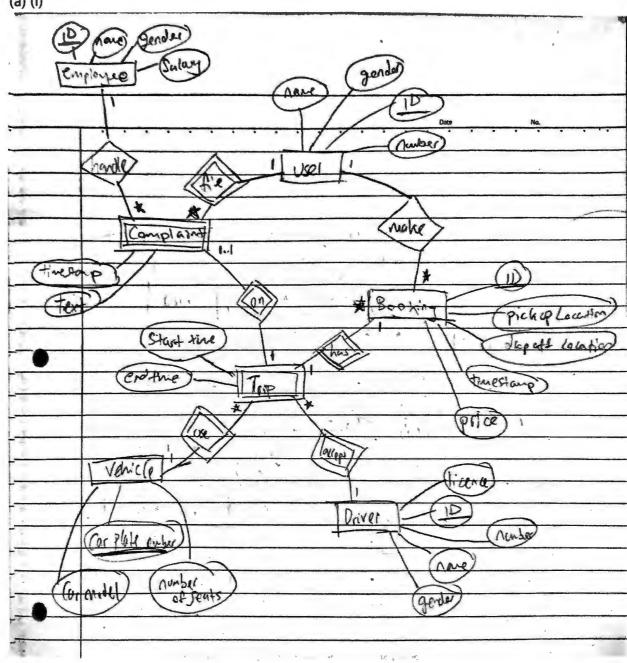
Solver: Shao Jie

Email Address: yews0012@e.ntu.edu.sg

1. (a) (i)



(ii).

User (id, name, gender, number)

Booking (id, pickup_location, dropoff_location, timestamp, price, user_id)

Driver (id, license, number, name, gender)

Vehicle (car plate number, car_model, number_of_seats)

Trip (booking id, driver id, vehicle car plate number, start_time, end_time)

Employee (id, name, gender, salary)

Complaint (booking id, driver id, vehicle car plate number, user id, timestamp, text, handle_by_employee_id)

*foreign key are in italic font, primary key are in bolded font.

: =	7
	TI CID O COURT CID MATCH
215	TI CID O COURT CID MATCH
	TI CID O MAICH
	Count Zno Al
1	
1	KI: = Y
	By fact (19) -> routh by B
	RI = Y COUNT (PI) -> POUNT, PI & PINNING = TEUR MATCH R3 := PIAYER MADERIAL MATCH R3 := PIAYER MADERIAL MADERIAL MATCH
	RA:= PLAYER MADERIAL M PID=D2 R)
	D. MAIDERIN M SINGE DO
	BA:= IT PID. (count) + (ount 2) -> Marchalon 93
	(continued) > Warmon 33
(iii)	
farr	RI = Tun Granici COURT
fart	RI = TI CIO O TYPE = GIOSS' COURT
(mr)	RI = TI CIO O TYPE = Grass, COURT RZ = MATCH M RI
(m.)	HE- MAICH M RI
fari	R3:= 8(TIP) O PUNION = FOUR R2)
fmr	HE- MAICH M RI
fmr	R3:= 8(TI P) O PIWIN = FOIL R2) RA := 8 (TI P) O PIWIN = TRUE R2)
fmr	R3:= 8(TI P) O PIWIN = FOIL R2) RA := 8 (TI P) O PIWIN = TRUE R2)
(m)	R3:= 8(TI P) O PIWIN = FOUR R2) RA:= 8 (TI P) O PIWIN = FOUR R2) RS:= PLAYER & PIP= PI R3
	R3:= 8(TI P) O PIWIN = FOIL R2) RA:= 8 (TI P) O PIWIN = FOIL R2) RS:= PLAYER M. PIP= P) R3 R6:= R5 M L PIP= P2 R4
	R3:= 8(TI P) O PIWIN = FOUR R2) RA:= 8 (TI P) O PIWIN = FOUR R2) RS:= PLAYER & PIP= PI R3
	R3:= 8(TI P) O PIWIN = FOIL R2) RA:= 8 (TI P) O PIWIN = FOIL R2) RS:= PLAYER D, PID= PI R3 R6:= RS D, PID: P2 R4 A7:= TI PID, Nove, Gooder, 208 O PI is NULL AND P2 is I
(CV)	R3:= 8(TI P) O PIWIN = FOIL R2) RA:= 8(TI P) O PIWIN = FOIL R2) RS:= PLAYER D. PIP= PI R3 P6:= RS D. PIP: P2 R4 A7:= TI PID. Nov., Gooder, 208 O PI is NULL AS P2 is I
	R3:= 8(TI P) O NWM:= FOIL R2) RA:= 8(TI P) O NWM:= FOIL R2) RS:= PLAYER DL PID= P) R3 RS:= RS DL PID= P2 R4 A7:= TI PID, Now, Gooder, 208 O P1 is NULL RD P2 is 1 A1:= Y (cont(P1) -> food, P1 O P1 bins= rive MATCH X (202) > Good P2 O Plaint = Foice MATCH
	R3:= 8(TIP) O PIWIN = FOIL R2) RA:= 8(TIP) O PIWIN = FOIL R2) RS:= PLAYER DL PIP= PI R3 R6:= RS DL PIP: P2 R4 A7:= TI PID, Nove, Gooder, DB O PI is NULL RO P2 is 1 A1:= Y (cont(P1) -> Cont, P1 O PIWINS = TWE MATCH R2:= Y (cont(P1) -> Cont, P2 O PIWINS = TWE MATCH
	R3:= 8(TI P) O PIWIN = FOIL R2) RA := 8 (TI P) O PIWIN = FOIL R2) RS := PLAYER D PID= PI R3 RS := RS D PID= PI R3 RS := TI PID Now, Gooder, DB O PI IS NULL AND P2 IS I A1 := Y Cont(P1) -> Cont, P1 O PIWINS = TWE MATCH R2 := Y Cont(P2) -> Cont, P2 O PIWIN = FOICE NATCH R3 := PLAYER N PID= PI R1 N PID= P2 R2 R3 := PLAYER N PID= PI R1 N PID= P2 R2 R4 := Y Cont(P2) -> Cont P2 O PIWIN = FOICE NATCH R5 := PLAYER N PID= PI R1 N PID= P2 R2 R6 := PLAYER N PID= PI R1 N PID= P2 R2
	R3:= 8(TI P) O PIWIN = FOIL R2) RA := 8 (TI P) O PIWIN = FOIL R2) RS := PLAYER D PID= PI R3 RS := RS D PID= PI R3 RS := TI PID Now, Gooder, DB O PI IS NULL AND P2 IS I A1 := Y Cont(P1) -> Cont, P1 O PIWINS = TWE MATCH R2 := Y Cont(P2) -> Cont, P2 O PIWIN = FOICE NATCH R3 := PLAYER N PID= PI R1 N PID= P2 R2 R3 := PLAYER N PID= PI R1 N PID= P2 R2 R4 := Y Cont(P2) -> Cont P2 O PIWIN = FOICE NATCH R5 := PLAYER N PID= PI R1 N PID= P2 R2 R6 := PLAYER N PID= PI R1 N PID= P2 R2
	R3:= 8(TIP) & PIWIN = FOLK R2) R5:= PLAYER D. PID= PIR3 R5:= RS D. PID: P2 R4 A7:= TIPID, Now, Gooder, DB B PI IS NULL AS P2 IS I A1:= Y (cont(P1) -> Cont, P1 O PIWINS = TWE MATCH R2:= Y (cont(P2) -> Cont, P2 B Plush = Folk NATCH R3:= PLAYER M PID = PIR1 MPD= P2 R2 R4:= TIPID, ((cont) + (cont2) -> Modernian R3 R4:= TIPID, ((cont) + (cont2) -> Modernian R3
	R3:= 8(TI P) O PIWING = FOLD, R2) R5:= PLAYER DL PID= PI R3 R5:= R5 DL PID= PI R3 R7:= TT PID, Nove, Gooder, DB O PI is NULL FOO P2 is 1 R7:= Y (cont(P1) -> (cont, P1 O PIWING = TIME MATCH R2:= Y (cont(P2) -> (cont, P2 O PIWING = TIME MATCH R3:= PLAYER N PID= PI R1 N PID= P2 R2 R4:= TI PID, ((cont) + (cont2) -> MODICINEN R3 R5:= TI monthwon -> monthwoon 2 R4 R5:= TI monthwon -> monthwoon R5
	R3:= 8(TIP) O PIWIN = FOIL R2) R1:= 8 (TIP) O PIWIN = FOIL R2) R5:= PLAYER DL PIP= PIR3 R5:= R5 DL PIP= PIR3 R7:= TIPID, Now, Gooder, 208 O PI is NULL RD P2 is I R1:= Y (cont(PI) -> (cont, P2 O Plains = Time MATCH R2:= Y (cont(PI) -> (cont, P2 O Plains = Time MATCH R3:= PLAYER N PID=PIR1 NPD=PIR1 R4:= Tipip, ((cont) + (cont2) -> Motinion R3 R4:= Timatchinan -> motinion R5

2. (a)

R (A, B, C, D, E)

Given input functional dependencies: $E \rightarrow C,D$; $A,B \rightarrow C$; $A,D \rightarrow B$; $B,D \rightarrow C$; $B,D \rightarrow E$;

Calculating attribute closures:

$$\{A\} + = \{A\}$$

$$\{B\}+=\{B\}$$

$$\{C\} + = \{C\}$$

$$\{D\}+=\{D\}$$

$$\{E\}+=\{C,D,E\}$$

$$\{A, B\} + = \{A, B, C\}$$

$${A, C} + = {A, C}$$

$$\{B, C\} + = \{B, C\}$$

$$\{A, D\}$$
+ = $\{A, B, C, D, E\}$ <--- Composite minimum candidate key

$$\{B, D\} + = \{B, C, D, E\}$$

$$\{C, D\} + = \{C, D\}$$

$$\{A, E\}+=\{A, B, C, D, E\} < --- Composite minimum candidate key$$

$$\{B, E\} + = \{B, C, D, E\}$$

$$\{C, E\} + = \{C, D, E\}$$

$$\{D, E\} + = \{C, D, E\}$$

$$\{A, B, C\} + = \{A, B, C\}$$

$$\{A, B, D\} + = \{A, B, C, D, E\} < ---$$
Superkey

$${A, C, D} + = {A, B, C, D, E} < --- Superkey$$

$$\{B, C, D\} + = \{B, C, D, E\}$$

$${A, B, E} + = {A, B, C, D, E} < --- Superkey$$

$${A, C, E} + = {A, B, C, D, E} < --- Superkey$$

$$\{B, C, E\} + = \{B, C, D, E\}$$

$$\{A, D, E\} + = \{A, B, C, D, E\} < ---$$
Superkey

$$\{B, D, E\} + = \{B, C, D, E\}$$

$$\{C, D, E\} + = \{C, D, E\}$$

$${A, B, C, D} + = {A, B, C, D, E} < --- Superkey$$

$${A, B, C, E} + = {A, B, C, D, E} < --- Superkey$$

$$\{A, B, D, E\} + = \{A, B, C, D, E\} < --- Superkey$$

$${A, C, D, E} + = {A, B, C, D, E} < --- Superkey$$

$$\{B, C, D, E\} + = \{B, C, D, E\}$$

$$\{A, B, C, D, E\} + = \{A, B, C, D, E\} < --- Superkey$$

Keys: {A, D}, {A, E}

Input relation is not in BCNF: it is not in 3NF and not all functional dependencies satisfy at least one of the following conditions:

- (1) The right-hand side is a subset of the left hand side, or
- (2) the left-hand side is a super key (or minimum key) of the relation. The functional dependencies that failed are: $E \rightarrow C$; $E \rightarrow D$; A, $B \rightarrow C$; B, $D \rightarrow E$.

Decomposing input relation into BCNF relations using input relation and input functional dependencies as sources.

Initial set of decomposed BCNF relations:

 $R_0(C, D, E)$ having FD(s): $E \rightarrow C$; $E \rightarrow D$.

R_1(A, B, E) having FD(s): (none).

 $R_2(A, B, C)$ having FD(s): $A,B \rightarrow C$

R_3_0(D, E) having FD(s): $E \rightarrow D$.

R_3_1(A, B, E) having FD(s): (none).

R 3 2 0(D, E) having FD(s): $E \rightarrow D$.

R_3_2_1(B, E) having FD(s): (none).

 $R_3_3(A, B, D)$ having FD(s): $A,D \rightarrow B$.

 $R_4_0(C, D, E)$ having FD(s): $E \rightarrow C$; $E \rightarrow D$.

R_4_1(B, E) having FD(s): (none).

R_5(A, B, D) having FD(s): A,D \rightarrow B.

Final set of decomposed BCNF relations (removing duplicate and subset relations):

R_O(C, D, E) having FD(s): $E \rightarrow C$; $E \rightarrow D$.

R_1(A, B, E) having FD(s): (none).

 $R_2(A, B, C)$ having FD(s): A,B \rightarrow C.

R 3 3(A, B, D) having FD(s): A,D \rightarrow B.

The following input functional dependencies were lost: $B,D \rightarrow C$; $B,D \rightarrow E$; $B,E \rightarrow D$.

(b)

Input relation is not in 3NF: it is not in 2NF and not all functional dependencies satisfy at least one of the following conditions:

- (1) The right-hand side is a subset of the left hand side,
- (2) the left-hand side is a super key (or minimum key) of the relation, or
- (3) the right-hand side is (or is a part of) some minimum key of the relation. The functional dependencies that failed are: $E \rightarrow C$; A, B $\rightarrow C$.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Decomposing input relation into 3NF relations using the Synthesis algorithm. For each functional dependency of the canonical cover set (merging functional dependencies having the same left-hand attribute(s)) of original relation's functional dependencies, create a relation schema with the attributes in that functional dependency (both sides). Checking if at least one key can be found in at least one newly formed 3NF relation. Since key {A, D} is present in at least one of the new 3NF relations, no new relation was created. Testing if any relation includes all of the attributes found in another relation (and deleting the duplicate or smaller one). No new relations were removed. Finished decomposing input relation into 3NF relations:

RO(C, D, E) having FD(s): $E \rightarrow C$; $E \rightarrow D$.

R1(A, B, C) having FD(s): A,B \rightarrow C.

R2(A, B, D) having FD(s): A,D \rightarrow B.

R3(B, D, E) having FD(s): $E \rightarrow D$; $B,D \rightarrow E$.

- 3. (a)
 - (i) 1
 - (ii) 1

(d)

- (iii) 5 (EXCEPT ALL which returns all records from the first table which are not present in the second table, leaving the duplicates as is.)
- (b)
 SELECT DISTINCT t.name
 FROM TEAMS t
 INNER JOIN PLAYERS p ON t.teamid = p.teamid
 WHERE p.numgoals>5;
- (c)
 CREATE VIEW PromotionSummary AS
 SELECT category, MIN(price) as minprice, MAX(price) as maxprice
 FROM BOOKS
 WHERE promoted = True
- (i) UPDATE EMP
 SET salary = salary/2
 - WHERE salary = (SELECT MAX(salary) FROM EMP);
- (ii) SELECT e.name
 FROM EMP e
 WHERE NOT EXIST (SELECT *
 FROM DEPENDENT d
 WHERE d.essn = e.ssn)

```
4. (a)
CHECK (
NOT EXIST(
SELECT *
FROM PC p
WHERE p.model NOT IN (SELECT pd.model FROM PRODUCT pd)
)
```

(b)	
A	В
1	8
100	0
100	0

