CMPSC 431W - Homework 2.
Yanjun Chen yfc 5289
1. Declarative query language= describe what you want rather than how you compute it. Relational calculus query is declarative query language.
Procedural query language: used to represent execution plans Relational algebra is procedual query language.
2. Assume cid are unique. (1.) relational algebra:
Thome, Ename, branch_name, num_employee
((Branch M Fersonal Banker eid) DEmployee) Branch eid = Personal Banker eid)
TRC: $\{R \mid \exists e \in Employee, \exists p \in Personal Banker, \exists b \in Branch (e.eid = p.eid \land$
b. manager_eid = e. eid 1 R. eid = p. eid 1 R. fname = e. fname 1
R. Iname = e. Iname Λ R. mum_employee = b. mum_employee Λ R. branch_name = b. branch_name) $\frac{3}{2}$
(2.) relational algebra:
Thame, known (\(\nabla \) birthday > " ob/02/2001" \(\nabla \) income > 10000 (Customer))
TRC= SPIJBCCUSTUS (-1 inthodays" 1/2 / A P Anguage A
SRI = Pt Customer (p. birthday > "ob/02/2001" 1 P. fname = R. fname 1 R. Iname = P. Iname 1 P. birthday = R. birthday 1 P. Income > (00000)

(3.) relational algebra: Ncid, frame, Ename, birthday (\(\num_\)employee>>\$ (Customer M Duns M Account \(\ni\) Branch) M Branch) IRC: IR I I O & Owns, JA & Account, JB & Branch, JC & Customer (B. mum employee >>5 \wedge A. accro = D. accro \wedge A. branch_no = B. branch_no \wedge D. cid = C. cid \wedge R. cid = D. cid A R. frame = C. frame A R. Iname = C. Iname A R. bîrthday = C. birthday) (4) relational algebra: Traid (Ouns) - Traid (Vouns. cid & d. cid / ouns. accno = d. accno (Pd (Ouns) x Ouns)) TRC-FR | YAtours, #B& Duns (A. accno = B.accno A A. cid + B.cid A R.cid = B.cid) { (For partial) Duners that DO have joint account:

A acomo = B acomo A A cid + B cid. same account number, but not same cid (different person). # I considered "Pittsburgh West" as branch name. But I think it can also be street (5.) relational algebra: Noid (V branch_name = "Pittsburgh West" (Branch M Employee M Personal))
intersect -(V branch_name = "Harrisburg South" (Branch M Employee M Personal Banker))) TRC = same cid. (unique). P. P. should be same person & P. in Pitts.

RIJCE Customer, IPE Personal Banker, IPE Personal Banker,

IE, Employee, IE, E Employee, IBE Branch, IB, E Branch

(C. cid = P. cid A P. cid = P. cid A P. eid = Eleid A Elbranch no =

BI. branch no A BI. branch name - "Ditter and West A B. B1. branch_no \ B1. branch_name = "Pittsburgh West \ Pz.eid = Ez.eid \
Ez. branch_no = Bz. branch_no \ Bz. branch_name = "Harrisburgh South" \ \ R. cid = C. cid) 9 get the cids as required.

(b.) relational algebra:
branch_name (Tename = "Jackson" (Employee & Branch) ()
V salary < 20000 (Employee M Prranch))
TRC =
{RIJE. F. Employee J. F. F. Employee, J. B. F. Branch (F. branch no =
FRIJE, t Employee, JES Employee, JB t Branch (E, branch-no = E2. branch_no \ E1. [name = "Jackson" \ E2. salary < 20007 \
El Land de la Parade na A D based name V- B branch name)
[E1. branch_mo = B. branch_no A R. branch_name = B. branch_name)
work in the same branch. momatter if $E_1 = E_2$.
(7) relational algebra:
Total (Texn-type = "check" / [amount [> 10000 (Transaction & Duns)) absolute value.
absolute value.
TRC:
$R[\exists T \in Transaction, \exists D \in Ouns(T. txn_type = "check" \land T. accno = D. accno \land [T. amount > 10000 \land R. cid = D. cid)]$
0. accno / [T. amount > 10000 / R. cid = D. cid) }
absolute value
18) relationed alaska
(8.) relational algebra: Select all autome meet the birthday To I were birthday (VI) of (2011) (Cuttomes)
To fname, Ename, birthday (V birthday < "08/21/1950" (Customer) -
TA. income > B. income & PA (Vbirthday < "D8/21/1950" (Customer)) × all possible PB (Vbirthday < "D8/21/1950" (Customer))) combination of austomer
The thought to a state of the s
all possible ITB (V birthday < "08/21/1950" (Customer)))
combination of austomer

(8.) TRC =
FRI 3 C/E Customer, Y C2 & Customer (C1. birthday < "08/21/1950" 1
SR = C1 & Customer, Y C2 & Customer (C1. birthday < "08/21/1950" \\ C2. birthday < "08/21/1950" \\ R. fname = C1. fname \\ R. fname = C1. fname \\ R. fname = C1. fname \\ R. fname \
R. frame = C1. frame / R. Iname = C1. Iname /
K. Birthady = Ci. Birthady) Select all (by t)
R. birthday = Ci. birthday) } Select all (by t) Customer with required birthday.
(9.) relational algebra:
Tid, salary (V Employee. Salary > M. salary (Employee M (FM (Employee M Branch)
TRC = regular employee manager = weid = manager_eid Employee))) FR JE t Employee, JME Employee, JBG Branch (E. branch_no = M. branch_no N M. eid = B. manager_eid N E. salary > M. salary N R. eid = E. eid N R. salary = E. salary) }
ERIJE + Francisco J Mc Francoure J BC Branch (F branch no =
M. branch_no M. M. eid = B. manager_eid M. E. salary > M. salary A
R.eid = E.eid / R.salary = E.salary) }
get full into all possible combination
10. relational algebra: get real id. get id of "Habeinto"
get full into get full into get id. get id of "fake into". Thame, Chame, income (Tid (Customer) - Tid (Tid (Tid branch-no) (Sustamer X Branch)
that actual This book not Duns M Account) M (ustomer)
that actual account account account account TRC:
H^{\bullet}
IR ∃C + Customer, ∃D + Duns, ∃A + Account, ∀B + Branch
(C'.cid=0.cid \land D.accno = A.aceno \land A.branch_no = B.branch_no \land R.fname = C.fname \land R.Iname = C.lname \land R.income = C.income)
The forme - compound of the inventer - contracting

11. relational algebra:	
Tid (Customer) - Tid (Tid, branch_no)	
Customer X (\(\nabla_{\text{200000}} \) \(\text{branch} \) \) all possible combination \(\lambda_{\text{cid}}, \text{branch}_{\text{no}} \) \(\lambda_{\text{cid}}, \text{branch}_{\text{no}} \) \(\lambda_{\text{cid}}, \text{branch}_{\text{no}} \) \(\lambda_{\text{cid}}, \text{branch}_{\text{no}} \)	
(Neid, branch-no (Duns M Account))	
all cid + branch - no that do exist	
TRC=	
FRIJCE Customer, JDE Duns, JA & Account, YB& Branch (
[20000 < B. budget < 200000 → (C. cid = O. cid ∧ D. accno =	
A. $accno \land A. branch _no = B. branch _no) \land R. cid = C. cid)$	
A. action 17. Trades = 100	