Reference Solution to Homework 4

Please note that most of the questions have more than one correct solution.

1 (7.24)
ℑ means "script F"
a. $\pi_{\text{Order\#, Ship\_date}}(\sigma_{\text{Warehourse\# ='W2'}}(\text{SHIPMENT})$

b. TEMP $\leftarrow \pi_{\text{Order\#}}$ (ORDER $*\sigma_{\text{Cname =' Jose Lopez '}}$ (CUSTOMER))
$\pi_{\text{Order\#, Warehouse\#}}$ (TEMP * SHIPMENT)

c. R(CUSTNAME, #OFORDERS, AVG_ORDER_AMT) $\leftarrow {}_{\text{Cname}}\Im_{\text{COUNT Order\#, AVERAGE Ord\_Amt}}$ (ORDER * CUSTOMER)

d. $\pi_{\text{Order \#, Odate, Cust\#, Ord\_Amt}}$ ($\sigma_{\text{Ship\_date - Odate > 30}}$ (SHIPMENT*ORDER))

e. T1 $\leftarrow \pi_{\text{warehourse\#}}(\sigma_{\text{City='New York'}}(\text{WAREHOURSE}))$
$(\pi_{\text{warehourse\#}}(\text{SHIPMENT}))/\text{T1}$

2 (7.24)
a. {t.Order#, t.Ship_date | SHIPMENT(t) and t.Warehourse# ='W2'}

b. {t.Order#, t.Warehouse# | SHIPMENT(t) and (∃c, o: CUSTOMER(c) and c.Cname='Jose Lopez' and ORDER(o) and c.Cust# = o.Cust# and o.Order# =t.Order#) }

3(7.24)

```
create table CUSTOMER
(Cust#        char(3)          not null,
Cname         varchar(30),
City          varchar(20),
primary key (Cust#)
)

create table ORDERS
(Order#       char(3)          not null,
Odate         datetime,
Cust#         char(3) ,
Ord_Amt       float,
primary key (Order#),
foreign key (Cust#) references CUSTOMER(Cust#)
)

create table ITEM
(Item#        char(2)          not null,
Unit_price    float,
primary key (Item#)
)

create table ORDER_ITEM
(Order#       char(3)          not null,
Item#         char(2)          not null,
```

```
 Qty            integer,
 primary key (Order#, Item#),
 foreign key (Order#) references ORDERS (Order#),
 foreign key (Item#) references ITEM (Item#)
 )


create table WAREHOUSE
 (Warehouse#          char(2)        not null,
 City                varchar(20),
 primary key (Warehouse#)
 )

 create table SHIPMENT
 (Order#          char(3)         not null,
 Warehouse#        char(2)          not null,
 Ship_date         datetime,
 primary key (Order#,Warehouse#),
 foreign key (Order#) references ORDERS(Order#),
 foreign key (Warehouse#) references WAREHOUSE(Warehouse#)
 )

 insert into CUSTOMER(Cust#, Cname, City)
 values ("111",    "Jose Lopez", "New York"  )
 insert into CUSTOMER(Cust#, Cname, City)
 values ("222",    "Franklin Wong", "Raleigh"  )
 insert into CUSTOMER(Cust#, Cname, City)
 values ("333",    "Alicia Zelaya", "New York" )

 insert into ORDERS(Order#, Odate, Cust#,Ord_Amt)
 values ("001", "02/20/2002", "111", 30.00)
 insert into ORDERS(Order#, Odate, Cust#,Ord_Amt)
 values ("002", "02/18/2002",   "222", 10.00)
 insert into ORDERS(Order#, Odate, Cust#,Ord_Amt)
 values ("003", "01/10/2002",   "333", 30.00)

insert into ITEM(Item#, Unit_price)
 values ("01", 10.00   )
 insert into ITEM(Item#, Unit_price)
 values ("02", 10.00   )
 insert into ITEM(Item#, Unit_price)
 values ("03", 10.00   )

 insert into ORDER_ITEM(Order#, Item#, Qty)
 values ("001", "01",1 )
 insert into ORDER_ITEM(Order#, Item#, Qty)
 values ("001", "02",2 )
 insert into ORDER_ITEM(Order#, Item#, Qty)
 values ("002", "02", 1 )
 insert into ORDER_ITEM(Order#, Item#, Qty)
 values ("003", "03",3 )

insert into WAREHOUSE(Warehouse#,City)
```

```
values ("W1", "New York"   )
insert into WAREHOUSE(Warehouse#,City)
values ("W2", "New York"   )
insert into WAREHOUSE(Warehouse#,City)
values ("W3", "Raleigh"   )

insert into SHIPMENT(Order#, Warehouse#,Ship_date )
values ("001", "W1", "02/23/2002")
insert into SHIPMENT(Order#, Warehouse#,Ship_date )
values ("002", "W2", "02/23/2002")
insert into SHIPMENT(Order#, Warehouse#,Ship_date )
values ("003", "W3", "02/23/2002")
```

3(7.24)
a)

```
select Order#, Ship_date
from SHIPMENT
where Warehouse#='W2'
```

```
 Order# Ship_date
 ------ -----------
 002            Feb 23 2002 12:00AM

(1 row affected)
```

b)

```
select SHIPMENT.Order#, SHIPMENT.Warehouse#
from SHIPMENT, ORDERS, CUSTOMER
where SHIPMENT.Order#=ORDERS.Order# and CUSTOMER.Cname='Jose Lopez'
and ORDERS.Cust#=CUSTOMER.Cust#
```

```
 Order# Warehouse#
 ------ ----------
 001    W1

(1 row affected)
```

c)

```
select CUSTOMER.Cname as CUSTNAME, COUNT(ORDERS.Order#) as #OFORDERS,
AVG(ORDERS.Ord_Amt) as AVG_ORDER_AMT
from CUSTOMER, ORDERS
where CUSTOMER.Cust#=ORDERS.Cust#
group by CUSTOMER.Cname
```

```
 CUSTNAME                    #OFORDERS   AVG_ORDER_AMT
 --------------------------- ----------- --------------------
 Alicia Zelaya                         1          30.000000
 Franklin Wong                           1          10.000000
 Jose Lopez                            1          30.000000

(3 rows affected)
```

d)

```
select ORDERS.Order#, ORDERS.Odate, ORDERS.Cust#, ORDERS.Ord_Amt
from ORDERS, SHIPMENT
where ORDERS.Order#=SHIPMENT.Order# and ( datediff( yy, ORDERS.Odate,
SHIPMENT.Ship_date )*365 + datediff( mm, ORDERS.Odate, SHIPMENT.Ship_date )*30 +
datediff( dd, ORDERS.Odate, SHIPMENT.Ship_date ) ) > 30

Order# Odate                   Cust# Ord_Amt
------ ------------------------- ----- --------------------
003    Jan 10 2002 12:00AM 333          30.000000

(1 row affected)
```

e)

```
SELECT Order#
FROM ORDERS O
WHERE NOT EXISTS
        (SELECT *
         FROM WAREHOUSE W
         WHERE W.City = 'New York' and NOT EXISTS
                (SELECT *
                 FROM SHIPMENT S
                 WHERE S.Warehouse# = W.Warehouse# AND S.Order# = O.Order#))
```

4

7.25

Here, We assume that in TRIP relation, the attribute SSN is the SALESPERSON's SSN, and the attribute Trip_ID in EXPENSE refers to the attribute Trip_ID in TRIP relation. So, from above assumptions, we know:

In TRIP relation, SSN is the foreign key, reference the attribute SSN in SALESPERSON.

In EXPENSE relation, Trip_ID is the foreign key, reference the attribute Trip_ID in TRIP.

a. $T(Trip\_ID, TotalAmount) \leftarrow {}_{Trip\_ID}\Im_{SUM\ Amount}(EXPENSE)$

$TRIP * \pi_{Trip\_ID}(\sigma_{TotalAmount > 2000}(T))$

b. $\pi_{SSN}(\sigma_{To\_City = 'Honolulu'}(TRIP))$

c. $\Im_{SUM\ Amount}(EXPENSE * \sigma_{SSN = '234\text{-}56\text{-}7890'}(TRIP))$

7.26

Here, We assume that in ENROLL relation, the attribute SSN is the STUDENT's SSN, and the attribute Course# refers to the attribute Course# in COURSE relation. In BOOK_ADOPTION relation, the attribute Course# refers to the attribute Course# in COURSE relation, and the attribute Book_ISBN refers to the attribute Book_ISBN in TEXT relation.

So, from above assumptions, we know:

In ENROLL relation, SSN and Course# are the foreign keys. SSN references the attribute SSN in STUDENT, and Course# references the attribute Course# in COURSE.

In BOOK_ADOPTION relation, Course# and Book_ISBN are the foreign keys. Course# references the attribute Course# in COURSE, and Book_ISBN references the attribute Book_ISBN in TEXT.

a. $\Im_{COUNT\ Course\#}(\sigma_{Name= 'John\ Smith'}(STUDENT) * \sigma_{Quarter = 'W99'}(ENROLL))$

b. $TEMP1(Course\#, Book\_ISBN, Book\#) \leftarrow {}_{Course\#, Book\_ISBN}\Im_{COUNT\ Book\_ISBN}(BOOK\_ADOPTION * \sigma_{Dept = 'CS'}(COURSE))$

$TEMP2(Course\#, Count\_ISBN) \leftarrow {}_{Course\#}\Im_{COUNT\ Book\_ISBN}(TEMP)$

$\pi_{course\#, Book\_ISBN, Book\_Title}(TEMP1 * (\sigma_{Count\_ISBN > 2}(TEMP2)) * TEXT)$

c. π_{Dept} (COURSE) - π_{Dept} (COURSE * (BOOK_ADOPTION *$\sigma_{Publisher \neq 'BC\ Publishing'}$ (TEXT)))

7.28

Here, We assume that in OPTIONS relation, the attribute Serial-No refers to the attribute Serial-No in CAR relation. In SALES relation, the attribute Salesperson-id refers to the attribute Salesperson-id in SALESPERSON relation, and the attribute Serial-No refers to the attribute Serial-No in CAR relation. So, from above assumptions, we know:

In OPTIONS relation, Serial-No is the foreign key. Serial-No references the attribute Serial-No in CAR.

In SALES relation, Salesperson-id and Serial-No are the foreign keys. Salesperson-id references the attribute Salesperson-id in SALESPERSON, and Serial-No references the attribute Serial-No in CAR.

a. $\pi_{Serial\text{-}No,\ Manufacturer,\ Sale\text{-}price}$ (CAR*(SALES *$\sigma_{Name\ =\ 'Jane\ Doe'}$ (SALESPERSON)))

b. TEMP ← $\pi_{Serial\text{-}No,\ Model,\ Manufacturer,\ Price}$ (CAR* OPTIONS)
RESULT←$\pi_{Serial\text{-}No,\ Model}$ (CAR – TEMP)

c. SALESPERSON]><| $_{Salesperson\text{-}id=Salesperson\text{-}id}$ SALES
It means to list to all the salespersons with their sales records, even if they don't have any.

The LEFT OUTER JOIN operation keeps every tuple in the left relation R in R]><|S. If no matching tuple is found in S, then the attributes of S in the join result are filled with null values. Here we can give an example:

SALESPERSON:

Salesperson-id	Name	Phone
01	Bob Smith	111-1111
02	John Wong	222-2222
03	Linda Wallace	333-3333

SALES:

Salesperson-id	Serial-No	Date	Sale-price
01	123	20020220	1000.00
02	234	20020218	2000.00

Then, the result relation is:

RESULT:

Salesperson-id	Name	Phone	Serial-No	Date	Sale-price
01	Bob Smith	111-1111	123	20020220	1000.00
02	John Wong	222-2222	234	20020218	2000.00
03	Linda Wallace	333-3333	NULL	NULL	NULL

d. RESULT ← SALESPERSON - $\sigma_{Name\ =\ 'Jane\ Doe'}$ (SALESPERSON)
The above query will list Salesperson-id, Name, Phone of all the salespersons except 'Jane Doe'

5

7.25

a) Assume that Trip_ID → Account#.
{t | TRIP(t) and (∃e: EXPENSE(e) and e.Trip_ID= t.Trip_ID and e.Amount > 2000)}

b) {t.SSN | TRIP(t) and t.To_City = ‘Honolulu’)}

7.26
b) {c.Course#, t.Book_ISBN, t.BookTitle | COURSE(c) and TEXT(t) and (∃b1,b2,b3: BOOK_ADOPTION(b1) and BOOK_ADOPTION(b2) and BOOK_ADOPTION(b3) and b1.Book_ISBN ≠ b2.Book_ISBN and b1.Book_ISBN ≠ b3.Book_ISBN and b2.Book_ISBN ≠ b3.Book_ISBN and c.Course# = b1.Course# and c.Course# = b2.Course# and c.Course# = b3.Course# and c.Dept=’CS’ and (t.Book_ISBN=b1.Book_ISBN or t.Book_ISBN=b2.Book_ISBN or t.Book_ISBN=b3.Book_ISBN)) }

c) {c.Dept | COURSE(c) and (∀c’: COURSE(c’) and c’.Dept = c.Dept and (∀d: BOOK_ADOPTION(d) and (d.Course# = c’.Course# → (∃t: TEXT(t) and t.Book_ISBN = d.Book_ISBN and t.Publisher = ‘BC Publishing’)))}

7.28
a) {c.Serial-No, c.Manufacturer, t.Sale-price | CAR(c) and SALES(t) and (∃s: SALESPERSON(s) and s.Name =’Jane Doe’ and c.Serial-No = t.Serial-No and t.Salesperson-id = s.Salesperson-id) }

b) {c.Serial-No, c.Model | CAR(c) and (∀o: OPTIONS(o) and o.Serial-No ≠ c.Serial-No)}

c) Natural join is as following:
{s.Salesperson-id, s.Name, s.Phone, t.Serial-No, t.Date, t.Sale-price | SALESPERSON(s) and SALES(t) and s.Salesperson-id=t.Salesperson-id }

d) {t | SALESPERSON(t) and t.Name ≠ ’Jane Doe’}

6(7.25)
a)

```
select SSN, From_City, To_City, Departure_Date, Return_Date, TRIP.Trip_ID
from TRIP, EXPENSE
where EXPENSE.Trip_ID=TRIP.Trip_ID
group by SSN, From_City, To_City, Departure_Date, Return_Date, TRIP.Trip_ID
having SUM (Amount) > 2000
```

SSN	From_City	To_City	Departure_Date	Return_Date	Trip_ID
999983212	Los Angeles	Chicage	1999-02-10	1999-02-23	101
999983212	Los Angeles	New York	1999-05-21	1999-06-01	102
999983212	San Francisco	Honolulu	2000-10-09	2000-10-11	103
234567890	Chapel Hill	Raleigh	1999-04-01	1999-04-10	213
234567890	Raleigh	Honolulu	1999-12-31	1999-01-30	221
888776666	Honolulu	New York	1997-11-11	1997-12-12	301

(6 rows affected)

b)

```
select SSN
from TRIP
where To_City='Honolulu'

SSN
---------
999983212
234567890
```

(2 rows affected)

c)

```
select SUM(EXPENSE.Amount) as Total_Trip_Expenses
from TRIP, EXPENSE
where  TRIP.SSN='234567890' and TRIP.Trip_ID=EXPENSE.Trip_ID

Total_Trip_Expenses
-------------------
          206500
```

(1 row affected)

6.(7.26)

a)

```
 select COUNT(ENROLL.Course_No)
 from   ENROLL, STUDENT
 where  STUDENT.Name='John Smith' and STUDENT.SSN= ENROLL.SSN and
ENROLL.Quarter='W99'

-----------
        2
```

(1 row affected)

b)

```
select C.Course_No, T.Book_ISBN, T.Book_Title
from COURSE C, TEXT T, BOOK_ADOPTION B
where COURSE.Dept='CS' and B.Course_No = C.Course_No and B.BOOK_ISBN =
          T.BOOK_ISBN and C.Course_No  in
                    (select  Course_No
                    from BOOK_ADOPTION
                    group by Course_No
                    having count(distinct Book_ISBN)>2
                    )
```

c) SELECT DISTINCT Dept
FROM COURSE C1
WHERE NOT EXISTS
(SELECT *
FROM COURSE C2, BOOK_ADOPTION B, TEXT T
WHERE C2.Dept = C1.Dept and C2.Course_No = B.Course_No and
B.Book_ISBN = T.Book_ISBN and T.Publisher != 'BC Publishing')

6(7.28)
a)

```
select CAR.Serial_No, CAR.Manufacturer, SALES.Sale_price
from CAR, SALES, CARSALESPERSON
where CARSALESPERSON.Name='Jane Doe' and
CARSALESPERSON.Salesperson_id=SALES.Salesperson_id and
CAR.Serial_No=SALES.Serial_No
```

```
Serial_No Manufacturer                    Sale_price
--------- --------------------------------- -----------
012345682 Chevrolet                          13800
012345683 Chevrolet                          18300
012345685 Honda                             12500
012345688 Mazda                             26799

(4 rows affected)
```

b)

```
select CAR.Serial_No, CAR.Model
from CAR
where  CAR.Serial_No not in
      (select OPTIONS.Serial_No
      from OPTIONS
      )
```

```
Serial_No Model
--------- ----------------------------------
012345678 Integra
012345679 911
012345680 Century
012345683 Blazer LS
012345684 Accord EX
012345685 Civic DX
012345688 Millenia
012345689 Diablo 6.0

(8 rows affected)
```

c) left outer join:

```
select P.Salesperson_id, P.Name, P.Phone, S.Serial_No, S.Date, S.Sale_price
from CARSALESPERSON P, SALES S
where P.Salesperson_id=S.Salesperson_id OR NOT EXISTS
          (SELECT *
           FROM SALES S2
           WHERE S2.Salesperson_id = C.Salesperson_id)
```

The following is the natural join operation:

```
select CARSALESPERSON.Salesperson_id, CARSALESPERSON.Name,
CARSALESPERSON.Phone,
SALES.Serial_No, SALES.Date, SALES.Sale_price
from CARSALESPERSON, SALES
where CARSALESPERSON.Salesperson_id=SALES.Salesperson_id
```

```
 Salesperson_id Name                           Phone        Serial_No
       Date       Sale_price
 -------------- ---------------------------------------- ------------ ---------
      ---------- -----------
 001            John Smith                     919-829-0001 012345678
       2001-10-11       22200
 001            John Smith                     919-829-0001 012345680
       2002-01-05       25300
 002            Jane Doe                       919-829-8932 012345682
       2001-09-09       13800
 002            Jane Doe                       919-829-8932 012345683
       2001-09-10       18300
 002            Jane Doe                       919-829-8932 012345685
       2001-09-14       12500
 002            Jane Doe                       919-829-8932 012345688
       2001-09-23       26799
 003            Franklin Wong                    919-827-9088 012345689
       2002-02-07       290000

(7 rows affected)
```

d)

```
select *
from CARSALESPERSON
where CARSALESPERSON.Name!='Jane Doe'
```

```
 Salesperson_id Name                           Phone
 -------------- ---------------------------------------- ------------
 001            John Smith                     919-829-0001
 003            Franklin Wong                    919-827-9088

(2 rows affected)
```