Database Management Fall 2023 HW3

An-Che, Liang B11705009

Problem 1

(a)

The SQL query is as follows:

```sql
SELECT e.Fname,
    e.Lname,
    wo.Pno
FROM EMPLOYEE AS e
    JOIN WORKS_ON AS wo ON e.Ssn = wo.Essn
WHERE EXISTS (
        SELECT p.Pnumber
        FROM PROJECT AS p
        WHERE p.Pnumber = wo.Pno
            AND p.Dnum = 5
    );
```

(b)

The SQL query is as follows:

```sql
SELECT e.Fname, e.Lname
FROM EMPLOYEE AS e
    LEFT JOIN DEPENDENT AS d ON e.Ssn = d.Essn
WHERE d.Essn IS NULL;
```

Problem 2

(a)

The SQL query is as follows:

```sql
SELECT E.Ssn,
    DATE(E.Bdate, 'auto'),
    D.Dname,
    COUNT(DISTINCT L.Dlocation)
FROM EMPLOYEE AS E
    JOIN DEPARTMENT AS D ON D.Dnumber = E.Dno
    JOIN DEPT_LOCATIONS AS L ON L.Dnumber = D.Dnumber
GROUP BY D.Dname,
    E.Ssn
HAVING COUNT(DISTINCT L.Dlocation) > 2
```

Which return the data:

Ssn	E.Bdat	Dname	Lcount
123456789	1965-01-08	Research	3
333445555	1955-12-07	Research	3
453453453	1972-07-30	Research	3
666884444	1962-09-14	Research	3

(b)

The SQL query is as follows:

```sql
SELECT D.Dnumber,
    D.Dname,
    FLOOR(
        (
            julianday("2022-10-01", "auto") - julianday(S.Bdate, "auto")
        ) / 365
    ) as Age
FROM DEPARTMENT AS D
    JOIN EMPLOYEE S ON S.Ssn = D.Mgr_ssn
WHERE D.Dnumber IN(
        SELECT E.Dno
        FROM EMPLOYEE AS E
        GROUP BY E.Dno
```

```
        HAVING COUNT(*) > 1
    )
```

Which return the data:

Dnumber	Dname	Age
5	Research	66
4	Administration	81

(c)

The SQL query is as follows:

```
WITH T AS (
    SELECT *,
        DENSE_RANK() OVER (
            ORDER BY cnt Desc
        ) AS rnk
    FROM (
            SELECT Pnumber,
                Pname,
                COUNT(*) as cnt
            FROM PROJECT AS P
                JOIN WORKS_ON AS WO ON WO.Pno = P.Pnumber
            GROUP BY P.Pnumber
        )
)
SELECT Pnumber,
    Pname,
    cnt
FROM T
WHERE rnk = 2
```

Which return the data:

Pnumber	Pname	cnt
1	ProductX	2
3	ProductZ	2

(d)

The SQL query is as follows:

```
WITH cnt_rnk AS (
    SELECT *,
        DENSE_RANK() OVER (
            ORDER BY cnt Desc
        ) AS rnk
    FROM (
            SELECT Pnumber,
                COUNT(*) as cnt
            FROM PROJECT AS P
                JOIN WORKS_ON AS WO ON WO.Pno = P.Pnumber
            GROUP BY P.Pnumber
        )
),
dependent_check AS (
    SELECT E.Ssn,
        CASE
            WHEN D.Essn IS NULL THEN "false"
            ELSE "true"
        END AS flag
    FROM EMPLOYEE AS E
        LEFT JOIN DEPENDENT AS D ON E.Ssn = D.Essn
    GROUP BY E.Ssn
)
SELECT E.Ssn,
    E.Fname,
    E.Lname,
    D.Mgr_ssn,
    DC.flag as have_dependent
FROM EMPLOYEE AS E
    JOIN DEPARTMENT AS D ON D.Dnumber = E.Dno
    JOIN WORKS_ON AS WO ON WO.Essn = E.Ssn
    JOIN cnt_rnk AS CR ON WO.Pno == CR.Pnumber
    AND CR.rnk = 2
    JOIN dependent_check AS DC ON DC.Ssn == E.Ssn
```

Which return the data:

Ssn	Fname	Lname	Mgr_ssn	flag
123456789	John	Smith	333445555	true
333445555	Franklin	Wong	333445555	true
453453453	Joyce	English	333445555	false
666884444	Ramesh	Narayan	333445555	false

(e)

The SQL query is as follows:

```sql
SELECT E.Ssn,
    COUNT(WO.Hours) as work_projects,
    SUM(
        CASE
            WHEN WO.Hours IS NULL THEN 0
            ELSE WO.Hours
        END
    ) as work_hours,
    COUNT(DISTINCT P.Plocation) as project_locations
FROM EMPLOYEE AS E
    LEFT JOIN WORKS_ON AS WO ON WO.Essn = E.Ssn
    AND WO.Hours IS NOT NULL
    LEFT JOIN PROJECT AS P ON WO.Pno = P.Pnumber
GROUP BY E.Ssn
```

Which return the data:

Ssn	work_projects	work_hours	project_locations
123456789	2	40	2
333445555	4	40	3
453453453	2	40	2
666884444	1	40	1
888665555	0	0	0
987654321	2	35	2
987987987	2	40	1
999887777	2	40	1

(f)

The SQL query is as follows:

```sql
SELECT S.Ssn,
    COUNT(E.Super_ssn)
FROM EMPLOYEE AS E
    RIGHT JOIN EMPLOYEE AS S ON E.Super_ssn = S.Ssn
GROUP BY S.Ssn
```

Which return the data:

Ssn	**COUNT(E.Super_ssn)**
123456789	0
333445555	3
453453453	0
666884444	0
888665555	2
987654321	2
987987987	0
999887777	0

Problem 3

CUSTOMER

Column	Meaning	Data Type	Key	Constraint	Domain
CitizenID	Citizen ID	char(10)	PK	Not Null	

MEMBER

Column	Meaning	Data Type	Key	Constraint	Domain
CitizenID	Citizen ID	char(10)	PK FK:CUSTOMER(CitizenID)	Not Null	
Name	Name	varchar(12)		Not Null	
Birthday	Birthday	date		Not Null	
Phone_Number	Phone number	varchar(64)		Not Null	
Email	Email	varchar(64)			
Referential triggers		**On Delete**		**On Update**	
CitizenID: CUSTOMER(CitizenID)		Cascade		Cascade	

TRAIN

Column	Meaning	Data Type	Key	Constraint	Domain
TrainID	Train ID	varchar(10)	PK	Not Null	
Date_of_Starting_Service	Date of starting service	date		Not Null	

TRIP

Column	Meaning	Data Type	Key	Constraint	Domain
TripID	Trip ID	varchar(10)	PK	Not Null	
TrainID	Train ID	varchar(10)	FK: TRAIN(TrainID)	Not null	
Referential triggers		**On Delete**		**On Update**	
TrainID: TRAIN(TrainID)		Cascade		Cascade	

STATION

Column	Meaning	Data Type	Key	Constraint	Domain
StationID	Station ID	varchar(10)	PK	Not Null	
Station_Name	Station name	varchar(10)		Not null	
Postal_Code	Postal code	varchar(6)		Not null, Unique	
Address_String	Station address	varchar(30)		Not null, Unique	

PASS

Column	Meaning	Data Type	Key	Constraint	Domain
TripID	Trip ID	varchar(10)	PK, FK: TRIP(TripID)	Not Null	
StationID	Station ID	varchar(10)	PK, FK: STATION(StationID)	Not Null	
Arrive_Time	Arrival time	time		Not Null	
Depart_Time	Departure time	time		Not Null	
Referential triggers		**On Delete**		**On Update**	
TripID: TRIP(TripID)		Cascade		Cascade	
StationID: STATION(StationID)		Cascade		Cascade	

ROUTE

Column	Meaning	Data Type	Key	Constraint	Domain
Start_Station	Starting station	varchar(10)	PK, FK: STATION(StationID)	Not Null	
End_Station	Ending station	varchar(10)	PK, FK: STATION(StationID)	Not Null	
Car_Level	Car level	varchar(10)	PK	Not Null	
Price	Price	int		Not Null	
Referential triggers		**On Delete**		**On Update**	
Start_Station: STATION(StationID)		Cascade		Cascade	
End_Station: STATION(StationID)		Cascade		Cascade	

TICKET

Column	Meaning	Data Type	Key	Constraint	Domain
Ticket_ID	Ticket ID	varchar(20)	PK	Not Null	
Booking_Time	Booking time	date		Not Null	
Payment_Time	Payment time	date			
Travel_Date	Travel date	date		Not Null	
Car_Level	Car level	varchar(10)		Not Null	
TripID	Trip ID	varchar(10)	FK: TRIP(TripID)	Not Null	
From_Station	From station	varchar(10)	FK: STATION(StationID)	Not Null	
To_Station	To station	varchar(10)	FK: STATION(StationID)	Not Null	
CitizenID	Citizen ID	varchar(10)	FK: MEMBER(CitizenID)	Not Null	
Status	Ticket status	varchar(15)		Not Null	{ paid, p+cancel, unpaid, unp+canceled}
Referential triggers		**On Delete**		**On Update**	
TripID: TRIP(TripID)		Cascade		Cascade	
From_Station: STATION(StationID)		Cascade		Cascade	
To_Station: STATION(StationID)		Cascade		Cascade	
CitizenID: MEMBER(CitizenID)		Cascade		Cascade	

Problem 4

(a)

We can normalize the original schema into the following:

TRIP(TripID, EmployeeID, StartDate, EndDate)
VISIT_Date(TripID, Date)
VISIT_CUSTOMER(TripID, CustomerID)
VISIT_PRODUCT(TripID, ProductID)

My design satisfy the following constraints:

1. 1NF, Since I eliminate multi-valued attributes.
2. 2NF, Since I create separate tables for sets of values that apply to multiple records.
3. 3NF, Since there are no non key attribute is transitively dependent on the primary key.
4. BCNF, Since for all functional dependency say $P \rightarrow Q$, P should is the primary key.
5. 4NF, Since there are no non-trivial multivalued dependencies other than a candidate key.

(b)

The relation schema is not 2NF, since the for the same **ServiceType**, **Fee** will be the same, thus does not satisfy the condition of 2NF. It should be normalized into:

Consulting(EngineerID, CustomerID, ConsultingDate, ServiceID)
Service(ServiceID, ServiceType, Fee)

The result is also 3NF, Since there are no non key attribute is transitively dependent on the primary key.

(c)

Product(ProductID, ProductName, ProductHeight, ProductWidth, ProductDepth)
Supplier(SupplierID)
ImportFrom(SupplierID, ProductID, Price)

My design satisfy the following constraints:

1. 1NF, Since I eliminate multi-valued attributes.
2. 2NF, Since I create separate tables for sets of values that apply to multiple records.
3. 3NF, Since there are no non key attribute is transitively dependent on the primary key.
4. BCNF, Since for all functional dependency say $P \rightarrow Q$, P should is the primary key.
5. 4NF, Since there are no non-trivial multivalued dependencies other than a candidate key.