COMP-3150 Exam

Lucas Sarweh 110042658

Q1: a) Commutative Law - The property where a term inside brackets and outside brackets are interchangeable for an operation. Ex: $(a \times b) \times c = a \times (b \times c)$

b) Domain Integrity - Columns in a databace must maintain a specified data type. The values entered must have that data type.

c) Entity Integrity - Every Entity set must have a primary key, this being unique and required.

d) DISTINCT - key word that includes no duplicate columns.

e) Aggregation Functions - Functions that return one value, from one column.

f) Correlated Subquery - A subquery that uses things from the main query. It can access things outside the subquery.

g) The layer in software architecture that acts as the backend and can access file info.

h) Functional Dependency - This happens when one column in a data-set automatically can determine another.

Q2: The database has properties that make it different like permanent storage, atomicity, and other integrities that garentee the data stored is always stored in a certain manner.

Application-level- data processing transactions are not atomic and do not garentee domain integrity (casting). Also, databases are queried using declarative languges and application-level using imperative language.

Q3: a) $\pi_{Name}((\pi_{Id}(Person)) / \pi_{FatherId}(Person))$

$\cup\ (\pi_{Id}(Person) / \pi_{motherId}(Person)]))$

b) ~~π Name, PlaceofBirth (~~

~~ρ(sonname/Name, SonBirthPlace/PlaceofBirth)(π~~

$\pi_{S.name, S.PlaceofBirth}\ \sigma(S.FatherId=Id)(\rho S(Person))$

c) $\pi_{F.Name, m.Name}\ \sigma(F.Id = FatherId \text{ AND } m.Id = motherId)$
$(\rho M)(\rho F)(Person)$

d) $\pi_{name}(Person) / \pi_{name}\ \sigma(F.Id = FatherId \text{ AND }$
$m.Id = motherId))(\rho F)(\rho M)$
$(Person)$

Q4: a)

```
SELECT Body FROM Comment
WHERE (SELECT MAX(Date) FROM comment
        WHERE (SELECT MAX(Time) From comment))
```

b)

```
SELECT Id FROM comment
WHERE Date = '2022' AND Body = 'COMP3150'
     AND UserId = (SELECT Id From User
                          WHERE Name = 'Hossein')
```

c)

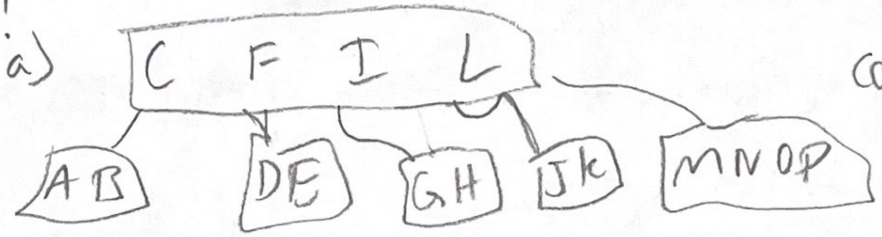
```
SELECT MAX(CNT) FROM
 (SELECT Id , COUNT(*) AS CNT
  FROM Like
WHERE CommentId = (SELECT Id FROM comment
                        WHERE UserId = (SELECT
                         Id FROM User
                        WHERE Name = 'Hossein'))
```

d)
```
SELECT Id FROM User
  GROUP BY Id
HAVING COUNT(*) = (SELECT MAX(CNT) FROM
  (SELECT Followee userId From Followership))
  ~~WHRE Id = IN (SELECT~~
```

e)
```
DELETE* FROM user WHERE
  Id = (SELECT userId FROM comment
        WHERE Id = NULL)
```

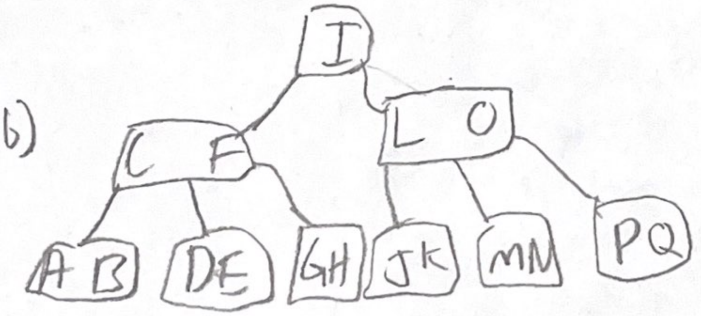
Q5!

a)



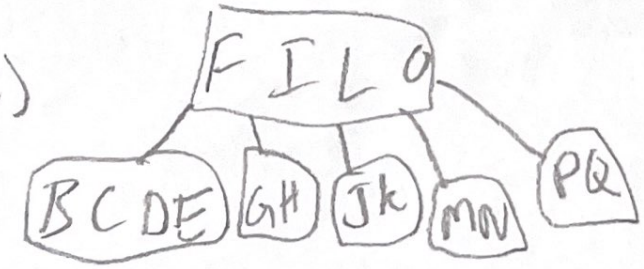
can't create new top row so add to end

b)



new Q allows balance for a new row

c)



Drop all before F to last row and bring down I

Q6: 1NF must have a key

Customer, Service → Phone, Address, Volunteer, Date, Donation

UNF unnormalized — T (Customer, Phone, Address, Volenteer,
Service, Date, Donation)

2NF

CK ~~Proper Subsets → Customer, Service, Donation,~~

Proper Subsets → Volunteer, Service, Date, Donation

non-key → Customer, Phone, address

Volunteer, Service, Date, Donation → Customer, Phone, address

3NF - No non-key determines anything

~~Ck1: Date, Donation~~ → Customer, Phone, address
~~Ck2: Service, Donation →~~ " "
Ck1: Volunter, Donation → " "
Ck2: Date, Service → " "
Ck3: Date, Volunteer → " "
Ck4: Service, Volunteer → " "
Ck5: Donation → " "

BCNF 3NF + left side superkeys only