ICCS240: Assignment 1

Natthakan Euaumpon
natthakaneuaumpon@gmail.com

February 2020

1: Warm-upQuestions

(1) How many relations over S,T are there?
2^{mn}

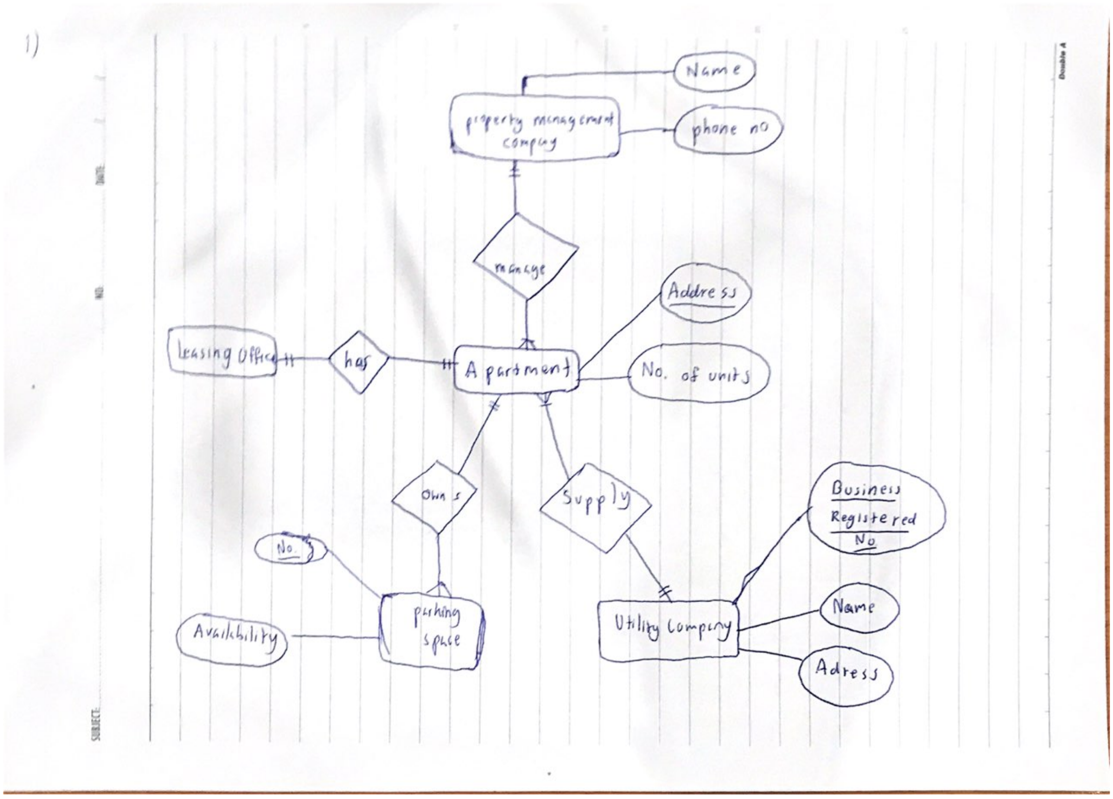
(2) How many functions from S to T are there
m^n

2: Relational Algebra

(1) $\Pi_B(R \bowtie S) = \Pi_B(R) \cap \Pi_B(S)$
By definition of natural join in set representation,
Let K be result set of the natural join between sets R ans S, T[A] be the tuples taken from table A and T[B] denotes the table taken from table B.
$K = R \bowtie S$
$K = \{T \in B_S B_R | T[S] \in s \wedge T[B_R] \in R\}$ We can project the result set, K
$\Pi_B(K) = \{T[Bs] \in S \wedge T[B_R] \in R\}$ We want to show that the intersection given yield the same result as the natural join.
$R \cap S = \{T | T \in RandT \in S\}$
We can project this as:
$\Pi_B[R] \cap \Pi_B[S] = \{T | T[B] ansT[B] \in S\}$

3: E/R Diagram and Relational Model

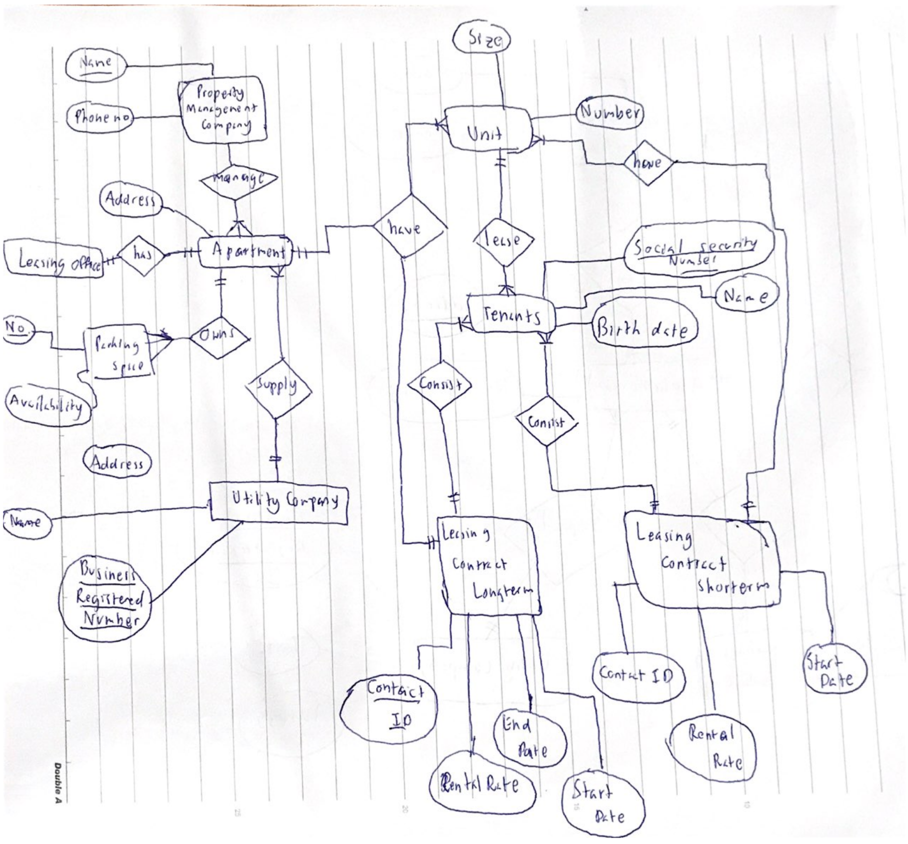
(1)E/R Diagram



(2) Relational Database Schema

The relational database schema :
PropertyManagementCompany(Name: string, Phone:int)
Apartment(Name: string, No.of unit: int, Leasing office:string, Apartment name:string)
Parkingspace(Number: int ,Availability: string)
UtilityCompany(Business Registered no.: int ,Name: string ,Address: string)

4: E/R Diagram



5: Relational Model and SQL

(1) Key and Foreign Key
BEER
Primary Key: brand
Foreign Key: none
COMPANY
Primary Key: HQ_location
Foreign Key: Brand
BAR
Primary Key: name
Foreign Key:Brand_of_beer_sold
SALE
Primary Key: none
Foreign Key:Brand_of_beer_sold and bar

(2) Relational Algebra Expression and SQL
(a)
SQL:

```
SELECT brand FROM BEER WHERE country_brewed != country_sold ;
```

Relational Algebra:

$\Pi_{brand}(\sigma_{country\_brewed!=country\_sold}(BEER))$
(b)
SQL:

```
SELECT SUM(number_of_sold)  FROM  SALE  GROUP BY year_record ;
```

Relational Algebra:
$\Pi_{SUM(number\_of\_sold)}(\sigma\gamma_{year\_record}(SALE))$
(c)
SQL:

```
SELECT brand, name from BAR, BEER where BAR.price_sold > BEER.standard_price ;
```

Relational Algebra:
$\Pi_{brand,name}(\sigma_{BAR.price\_sold>BEER.standard\_price}(BARXBEER))$

6: SQL

(1) Find the number of distinct makers for each type of computers

```
SELECT DISTINCT maker FROM Computer;
```

(b) Find the maker that produced the largest number of computers

```
SELECT * FROM
(SELECT maker FROM Computer ORDER BY (SELECT COUNT(model)
from Computer GROUP BY maker )) LIMIT 1;
```

(3) Without using a sub-query, find a list of PC.model-Laptop.model pairs whose difference in price is less than $100. Your list should also print the price difference between them

```
SELECT compc.model, comlaptop.model,
ABS(comps.price - comlaptop.price) difference FROM computer compc
INNER JOIN computer comlaptop ON compc.maker=comlaptop.maker
WHERE compc.type='pc' AND comlaptop.type='laptop' AND
ABS(compc.price - comlaptop.price)<100
```