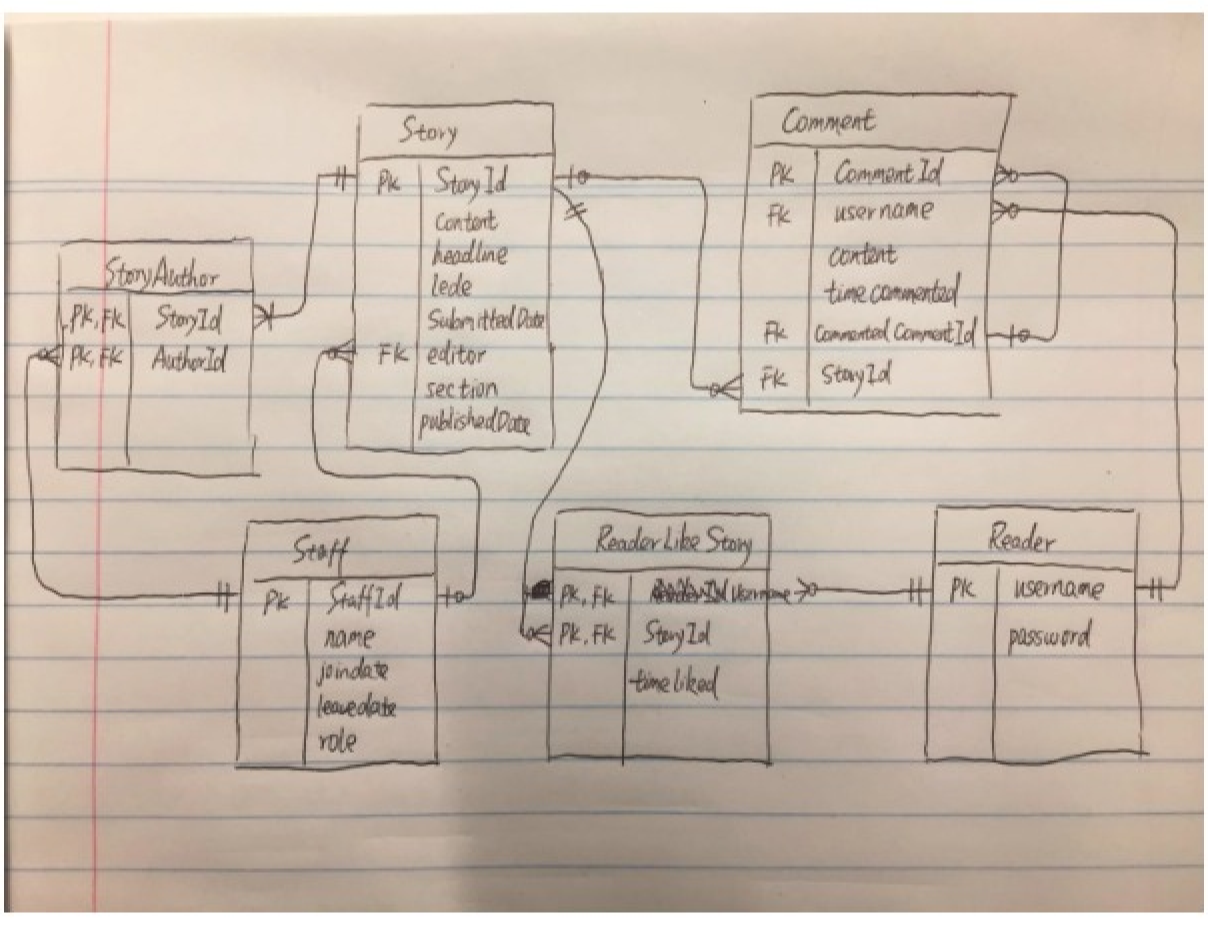
2016

Q1



Q2(unsure for the correctness)

A)

```
SELECT tenant AS TenantId, SUM(rent) AS TotalRent
FROM Tenancy
GROUP BY TenantId
ORDER BY TotalRent DESC
LIMIT 5;
```

B)

```
SELECT givenName
FROM Tenant
WHERE LENGTH(givenName) =
(SELECT MAX(LENGTH(givenName)) FROM Tenant);
```

C)

```
SELECT CONCAT(unitnum, ", ", propertynum, " ", street, ", ", suburb, ", ", postcode) AS Address
FROM Property
WHERE id IN
(SELECT id FROM House WHERE Bedrooms = 1
UNION
SELECT id FROM Apartment WHERE Bedrooms = 1);
```

D)

```
SELECT CONCAT(givenName, " ", surName) AS Name
FROM Tenancy INNER JOIN Tenant ON Tenancy.tenant = Tenant.id
INNER JOIN Property ON Tenancy.property = Property.id
GROUP BY Tenant.id
HAVING COUNT(DISTINCT Property.propertytype) > 1;
```

Q3

It has repeated groups.

tables(using italic representing foreign keys):

- R1(, givenName, surname)
- R2(*studentId*, *assignmentId*, mark, dataSubmitted)
- R3(, assignmentTitle)

Q4 ¶

- SMALLINT
- VARCHAR(15)
- CHAR(10)
- SMALLINT
- DATE
- CHAR
- CHAR(2)
- CHAR(4)
- SMALLINT
- TINYINT

Q5

A)

- It's a good fit for geographically distributed users.
- Having multiple server sites will be easier to handle large amount of data(good scalability).
- Users can access data faster from a local server.

B)

Replication: It means the same pieces of data are stored multiple times on different servers.

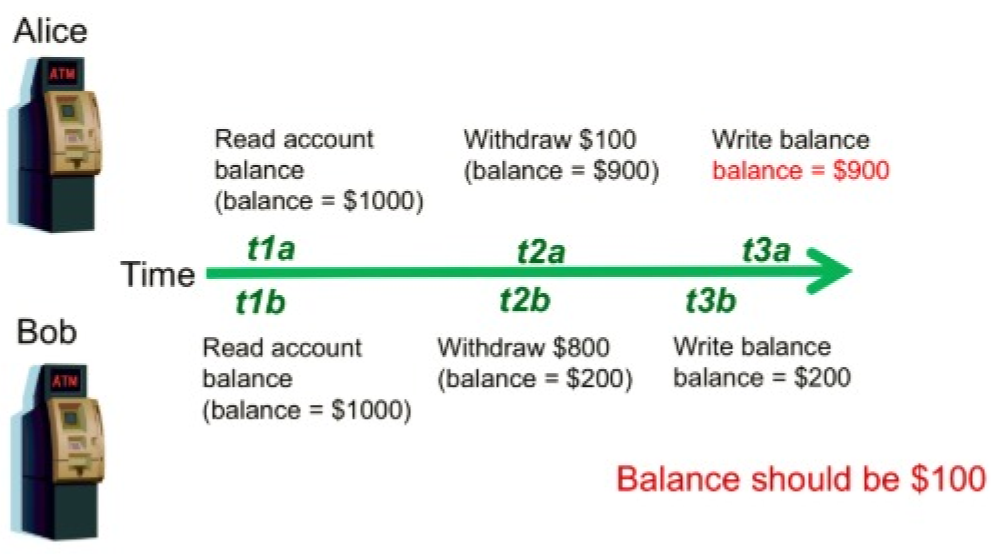
Partition: It means tables are split into rows(horizontal partitioning) or columns(vertical partitioning) and they are stored on different server sites. All of them form the entire logical database.

C)

- It adds complexity to management
- If there are multiple copies of data, it will increase the storage requirements
- Many server sites lead to high chance of being attacked
- It is more likely to have data consistency issue

Q6

A)


Alice
ATM
Read account balance (balance = $1000)
Withdraw $100 (balance = $900)
Write balance balance = $900
Time
t1a
t2a
t3a
t1b
t2b
t3b
Bob
ATM
Read account balance (balance = $1000)
Withdraw $800 (balance = $200)
Write balance balance = $200
Balance should be $100

B)

- It allows user to define a unit of work
- It allows concurrent access to the database

Q7

5000	5500	10000
0	1000	10000
3000	3100	4000
3	3	3

Q8

CAP theorem says that we can only have two out of consistency, partition tolerance and availability.

Say, we now have server sites A, B and C. Now, server site C lost its network connection. Any new updates happen in server site C will not be shown in A and B. If we want server site A and B to have data consistency, we will need to shut down all our servers(since the data in A is different with C, we don't want our user to see different results in A and C, the only thing we can do is to shut down all of them). If we choose availability, our user can still have the service, but they will have consistency issue.

Which way to go depends on the business rule and actual situation, if we are running a bank, we would want consistency. If we are running a social media, we would want availability.