Part A.

For the following SQL schema:

create table Building (

id int primary key,

address varchar(750) not null unique,

photo mediumblob

) engine=InnoDB;

create table Apartment (

id int primary key,

number varchar(31) not null,

building int not null references Building(id)

on update cascade on delete cascade,

photo mediumblob,

rent double,

unique(number, building)

) engine=InnoDB;

create table Room (

id int primary key,

area double not null,

type varchar(50),

apartment int not null references Apartment(id)

on update cascade on delete cascade,

photo mediumblob

) engine=InnoDB;

create table Person (

id int primary key,

name varchar(5000) not null,

email varchar(100) not null unique

) engine=InnoDB;

create table Owner (

person int references Person(id)

on update cascade on delete cascade,

apartment int references Apartment(id)

on update cascade on delete cascade,

primary key(person, apartment)

) engine=InnoDB;

Assume that: the following are the sizes, counts and times:

1. A record requires an average of 6 bytes administrative overhead, including the space required for specifying the rid.
2. An int uses 4 bytes.
3. A double uses 8 bytes.
4. A building address uses an average of 150 bytes.
5. An apartment number uses an average of 10 bytes.
6. A person name or email address uses an average of 25 bytes.
7. A room type uses an average of 10 bytes.
8. A photo uses an average of 32K bytes stored separately from the record. Within the record a photo uses an average of 10 bytes.
9. There are 20K buildings.
10. There are 2M apartments and 2M persons.
11. There are 20M rooms.
12. There are 20 room types.
13. On average each apartment is owned by 2 persons.
14. The disk block size is 8K = 8192 bytes.
15. A random disk access requires 10ms.
16. A page can be transferred (read or write) in 0.05 ms = 50 microseconds.

Show an optimized query evaluation plan for each of the following queries. Estimate the size of the data flow along each edge in the query evaluation plan. Assume that there is a B-tree index on the room type column.

1. Find all apartments, showing the address and apartment number that have 3 or more rooms with the same type.
2. select b.address, a.number
3. from Building b, Apartment a, Room r
4. where b.id = a.building
5. and r.apartment = a.id
6. group by b.id, a.id, r.type
7. having count(r.id) >= 3
8. List the buildings in order by address together with the number of apartments in each building.
9. select b.address, count(\*)
10. from Building b, Apartment a
11. where b.id = a.building
12. group by b.id
13. order by b.address
14. Find the distinct addresses of all buildings that have two different apartments owned by the same person who does not have email address fred@gmail.com.

Part B.

For the following schedules, determine whether each schedule is view serializable, conflict serializable and/or strict:

1. R1(X),R2(Y),W2(Y),W2(X),C2,W3(Y),C3,R1(X),C1
2. R1(X),R2(Y),W3(Z),W2(Y),W2(X),R1(Z),W3(Y),C3,W2(X),C2,C1
3. R1(X),R2(Y),R3(Z),W2(Y),W2(X),W1(Z),W3(Y),C3,W2(X),C2,C1