UNITS 2 AND 3

Consider the following relational schema about a Taxi company:

Taxi (*plate_num*: char(10), *age*: int, *km*: int, *capacity*: char(15))
PK: {*plate_num*}
NNV: {*km, capacity*, *age*}

Taxi_driver (*license_num*: char(10), *name*: char(50), *phone*: char(10), *address*: char(50), *age*: int)
PK: {*license_num*}
NNV: {*name*}

Workshop (*wk_id*: char(5), *name*: char(50), *zip_code* : char(5), *phone*: char(10), *supervisor*: char(50))
PK: {*wk_id*}
NNV: {*supervisor*}

Drives (*license_num*: char(10), *plate_num*: char(10), *ddate*: date, *start*: time, *end*: time)
PK: {*license_num*, *ddate*}
NNV: {*plate_num*}
FK: {*license_num*}→ Taxi_driver
FK: {*plate_num*}→ Taxi
NNV: {*start*}

Repair (*wk_id*: char(5), *plate_num*: char(10), *rdate*: date)
PK: {*plate_num*, *rdate*}
NNV:{*wk_id*}
FK: {*wk_id*}→ Workshop
FK: {*plate_num*}→ Taxi

Where the meaning of the attributes is the following:

Taxi : *plate_num*: Number plate of the car.
age: Age of the car.
km: The distance that the car has travelled.
capacity: Maximum number of passengers of the car.

Taxi_driver: *license_num*: Number of the driving license.
name: Name of the taxi driver.
age: Age of the taxi driver.
phone: telephone number of the taxi driver.
address: Address of the taxi driver.

Workshop: *wk_id*: Car workshop code.
name: Name of the car workshop.
zip_code: Zip code of zone where the workshop is located.
phone: Telephone number.
supervisor: Person who supervises the workshop.

Drives: The taxi driver with license number *license_num* drove (or is driving) the taxi with plate number *plate_num* the day d*date* from *start* to *end* hours.

Repair: The car workshop *wk_id* repaired the taxi with plate number *plate_num* the day *rdate.*

1) ¿What are the four properties of transaction? Enumerate the 4 ,and then define 2 (only 2) of them *(0.5 points)*

2) Write the following SQL queries:

   a) List the plate number and the capacity of the taxi that is more than 10 years old and has been used in most km (highest number of km). *(0.5 points)*

   b) List the plate number and the capacity of the taxis that haven't been repaired and have been driven by only one taxi driver. *(0.75 points)*

   c) List the code and name of the car workshop that has repaired most taxis in only one day. *(0.75 points)*

   d) List, for all the taxis in the database, the plate number, the age, the number of repairs performed on it, and the number of different taxi drivers who have driven the car. *(0.75 points)*

   e) List the plate number and the capacity of the taxis that have been repaired in all the workshops in the zone with zip code 46006 (if there is some workshop in that zone). *(0.75 points)*

1)

Atomicity:

A transaction is an indivisible unit that is either performed in its entirety or is not performed at all (“All or nothing”).

Consistency:

The transaction must transform the DB from one consistent state to another consistent state (all integrity constraints must be met)

Isolation:

Concurrent transactions execute independently: All the partial effects of incomplete transactions should not be visible to other transactions

Durability:

The effects of a successfully completed (committed) transaction are permanently recorded in the DB and must not be lost because of a subsequent system or other transaction failure

a)

```
SELECT T.plate_num, T.capacity
FROM Taxi T
WHERE T.age>10 AND km =  (SELECT MAX(T1.km)
                          FROM Taxi T1
                          WHERE T1.age>10) ;
```

b)

```
SELECT T.plate_num, T.capacity
FROM Taxi T
WHERE T.plate_num NOT IN (SELECT R.plate_num FROM Repair R)
  AND 1 = (SELECT COUNT(DISTINCT D.license_num)
           FROM Drives D
           WHERE D.plate_num=T.plate_num) ;
```

c)

```
SELECT T.wk_id, T.name
FROM Workshop T
WHERE T.wk_id IN (SELECT R.wk_id
                  FROM Repair R
                  GROUP BY R.wk_id, R.rdate
                  HAVING COUNT(R.plate_num) = (SELECT MAX(COUNT(R1.plate_num))
                                               FROM Repair R1
                                               GROUP BY R1.wk_id, R1.rdate);
```

d)

```
SELECT T.plate_num, T.age, COUNT(DISTINCT(R.rdate)),
         COUNT(DISTINCT(license_num)
FROM Taxi T LEFT JOIN Repair R ON T.plate_num = R.plate_num
             LEFT JOIN Drives D ON T.plate_num = D.plate_num
GROUP BY T.plate_num, T.age
```

e)

```
SELECT T.plate_num, T.capacity
FROM Taxi T
```

```
WHERE NOT EXISTS (SELECT *
                  FROM Worskhop W
                  WHERE W.zip_code=’46006’
                    AND NOT EXISTS(SELECT *
                                   FROM Repair R
                                   WHERE T.plate_num = R.plate_num
                                     AND R.wk_id = W.wk_id )
 AND EXISTS (SELECT *
             FROM Worskhop W1
             WHERE W1.zip_code = ’46006’);
```