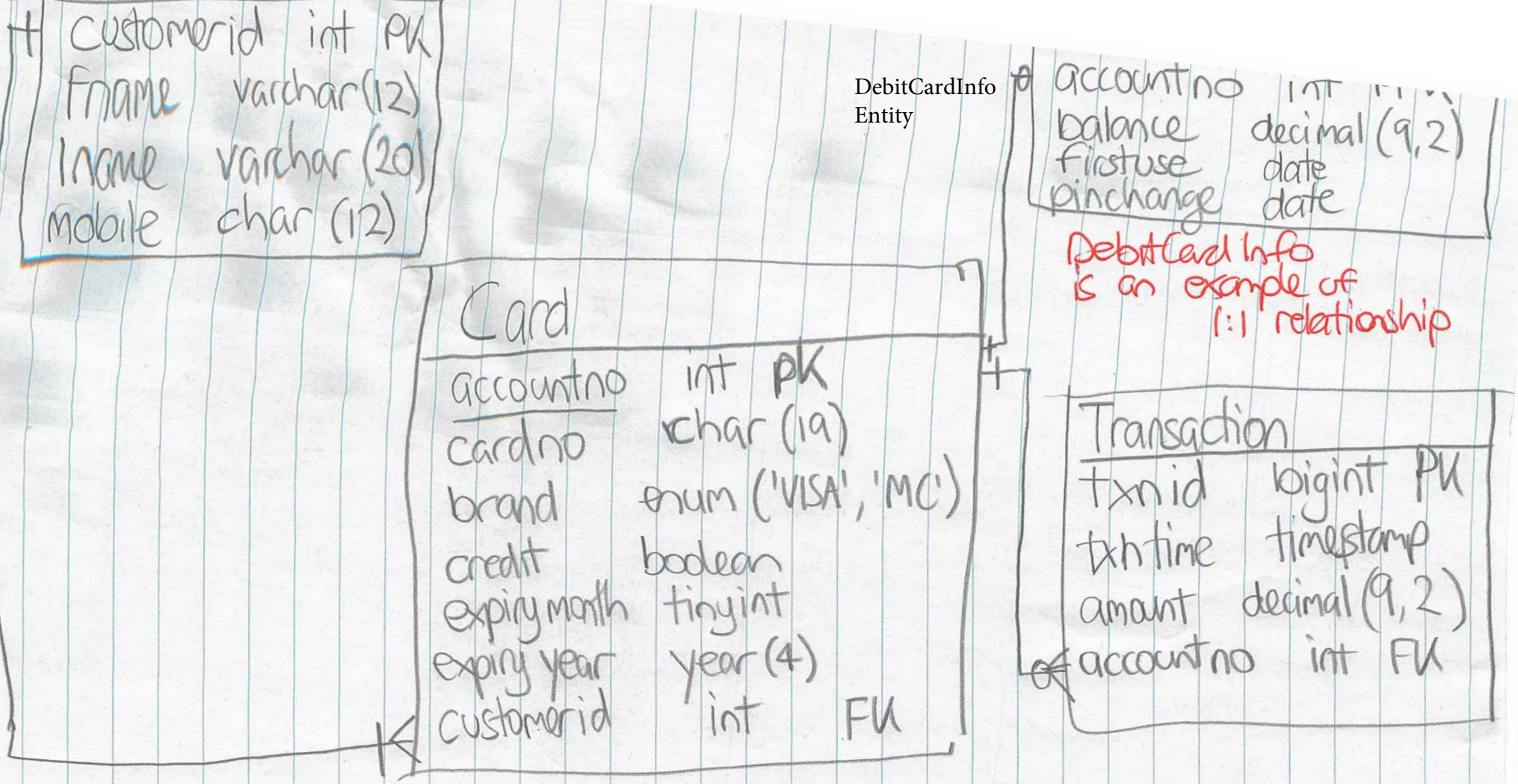


nb: card type (brand) could be a separate entity

expiry could be a char(5) eg 06/19 rather than 2 attributes

2

2.1
```
SELECT station.name
FROM station INNER JOIN line
ON station.lineid = line.lineid
WHERE line.name = 'WESTERN'
ORDER BY station.sequence;
```

nb must use inner join as station + line tables share 2 columns name and lineid!

2.2.
```
SELECT customer.username, COUNT(journey.journeyid),
       SUM(journey.cost)

FROM customer NATURAL JOIN journey
     customer
WHERE journey.faretype = 1

GROUP BY customer.username;
```

2.3
```
SELECT station.name
FROM station
WHERE station.stationid
NOT IN
      (SELECT journey.startstationid as stationid
       FROM journey
       UNION
       SELECT journey.endstationid as
              stationid
       FROM journey
      );
```

2.4.

```
SELECT  customer.username
FROM    customer natural join journey
        INNER JOIN station ON
        station.stationid = journey.startstationid
        OR
        station.stationid = journey.endstationID
GROUP BY customer.username
HAVING count(DISTINCT(station.lineid)) =
       (SELECT count(*)
        FROM line);
```

alternatively

```
SELECT customer.username
FROM customer
WHERE NOT EXISTS
   (SELECT *
    FROM line
    WHERE NOT EXISTS
       (SELECT *
        FROM journey INNER JOIN station
        ON station.stationid = journey.startstationID
        OR
         station.stationid = journey.endstationID
       )
   );
```

3

PET (PetID, Pet_name, type, Owner_id*)

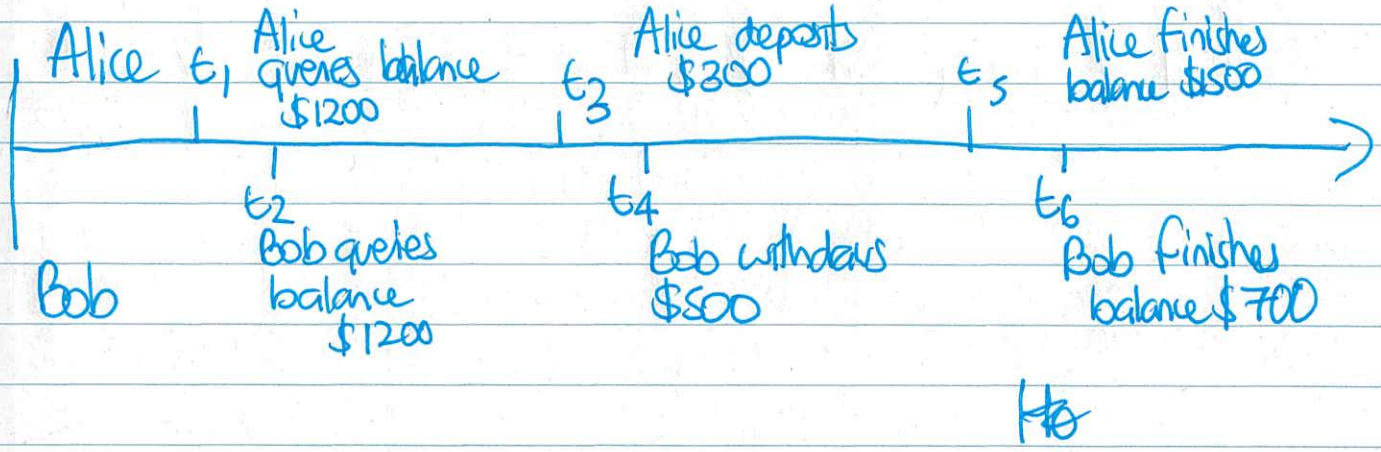
OWNER (OWNER_ID, OWNER_NAME, MOBILE)

PROCEEDURE (PROC_ID, PROC_DESC)

VISIT (VISIT, PROC_ID*, PET_ID*)

4.

The last update is where an update to a record in the database is lost due to another transaction overwriting the record.



At time t_6 the actual balance is

$$1200^{t_1,t_2} + 300^{t_3} - 800^{t_4} = \$1000$$

the transaction details are lost.

5.

ACID Properties:

Atomic - ~~occur~~ all changes succeed or all ~~change~~ fail

Consistent - what was true before the transaction is true after.
eg: an update to a not null column must still contain a value after the transaction

I ~~Independent - transactions are independent~~
Isolation - transactions must run in isolation changes are not visible to other sessions

D - durable once the change is made it is durable to all users.

However ~~Base~~ BASE (Basically Available, Soft state Eventually consistent).

NoSQL does not need to focus on consistency, isolation or durable changes.
All changes are eventually consistent at a future date. Changes do not need to occur in isolation.
NoSQL focuses on availability of the database.
Relational databases focus on the consistency of data