Part 1

Since multiple people are trying to access, update and update product inventory at the same time it may show values that are incorrect.

It may lead to data inconsistency; data lost and maybe even slow the database searching speed.

Dirty reads occur on user B because he is attempting to purchase at the same time when while user A is purchasing the product. Non repeatable reads problem occurs on user A who attempts to purchase a product, reducing its inventory count. Phantom reads problem occurs on user B because he is attempting to purchase while user A is already purchased the product.

Part 2

1.

```
LOCK TABLE tyontekija WRITE;
SELECT * FROM tyontekija WHERE sukunimi = "Ahola";
UNLOCK TABLES;
```

Prevents multiple transactions from updating the same data simultaneously.

2. MVCC creates duplicate copies of records so that data can be safely read and updated at the same time.

```
SELECT palkka FROM tyontekija WHERE sukunimi = "Ahola" FOR update;
```

```
START TRANSACTION;
UPDATE tili SET Saldo = Saldo - 100;
COMMIT;
```

3.

Old transactions get priority over new ones. This method ensures that transactions are executed in a consistent order, preventing conflicts.

Questions

Question1

prevents multiple transactions from updating the same data simultaneously. MVCC creates duplicate copies of records so that data can be safely read and updated at the same time. Older transactions get priority over new ones. This method ensures that transactions are executed in a consistent order, preventing conflicts.

Question 2

I recommend lock method because others can't interfere while you are doing your query

Part 3

```
CREATE VIEW customer_view
SELECT productname, amount, timestmp
FROM products;
```

view allows up to date scene. Views don't show data's that are locked or haven't been committed. No matter what the data shows is always up to date.

The trade-offs are that you can't build an index on the view.