function add ( )
{

Named fxn

}

Const addn = add ( )

Arrow fxn :-

(i) ( ) => expression

(ii) (a, b) => expression

(iii) ( ) => {

- - - - .

- - - - - .

- - - - - - `

}

function FuncName (arg1, arg2)
{

Named

}

FxnName (Params, Params)

const add = function (arg1, arg2) {

};

"Anonymous fxn"

add(Rooms, Runs);

Callbacks

add(a, b)

add(a, b, fxn);

Callback

function Calculator(a, b, callback) {

}

fxn

fxn

return callback(a, b);

callback → call

fxn

(1)

Fxn

```
function add(a, b)
{
    return a+b;
}

Var res = Calculator(3, 5, add)

Console.log(`${res}`);
```

Fxn as Param Pass

Call

"Asynchronous"

Callback

setTimeout(() => {

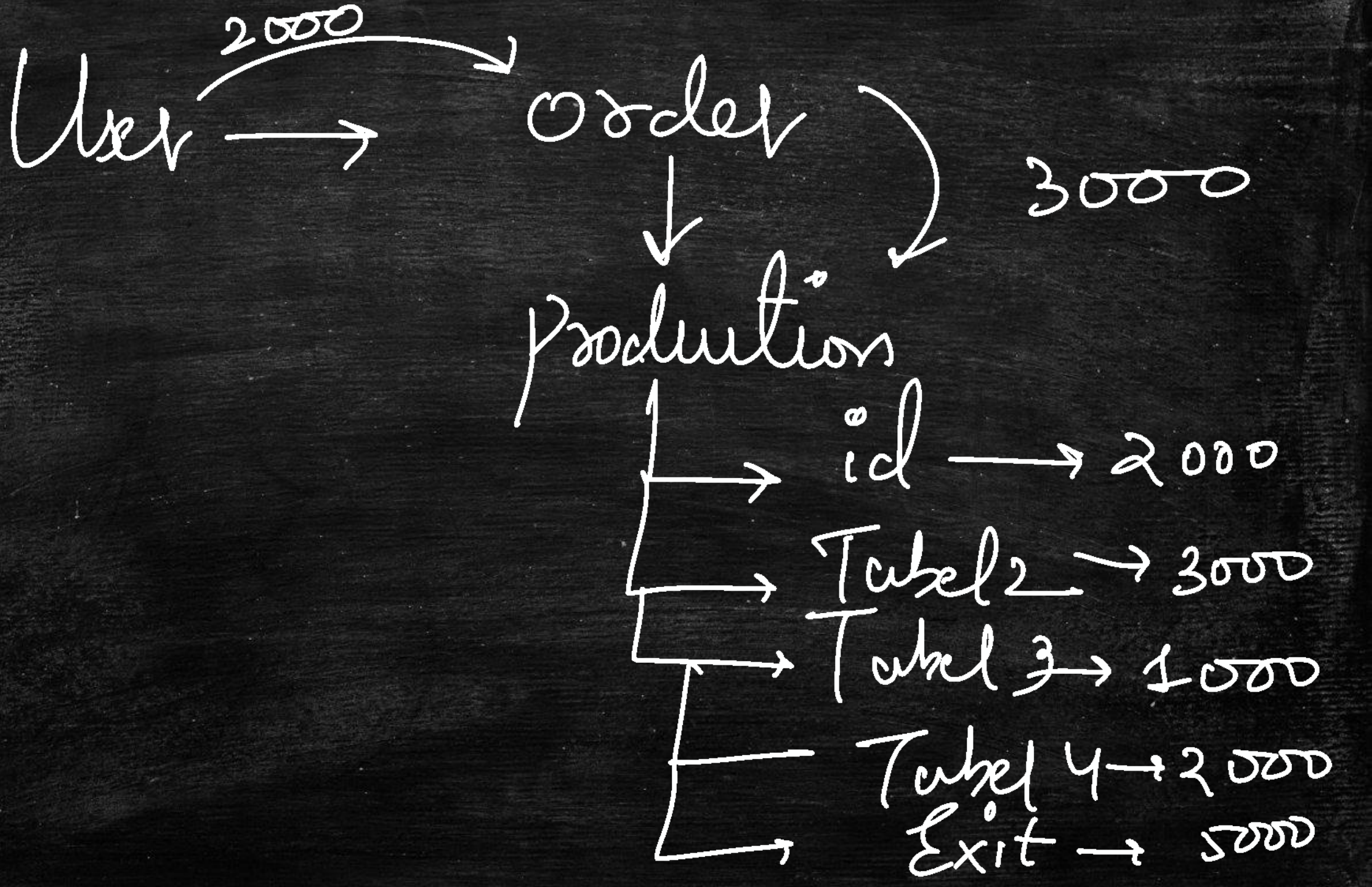
}, Time)

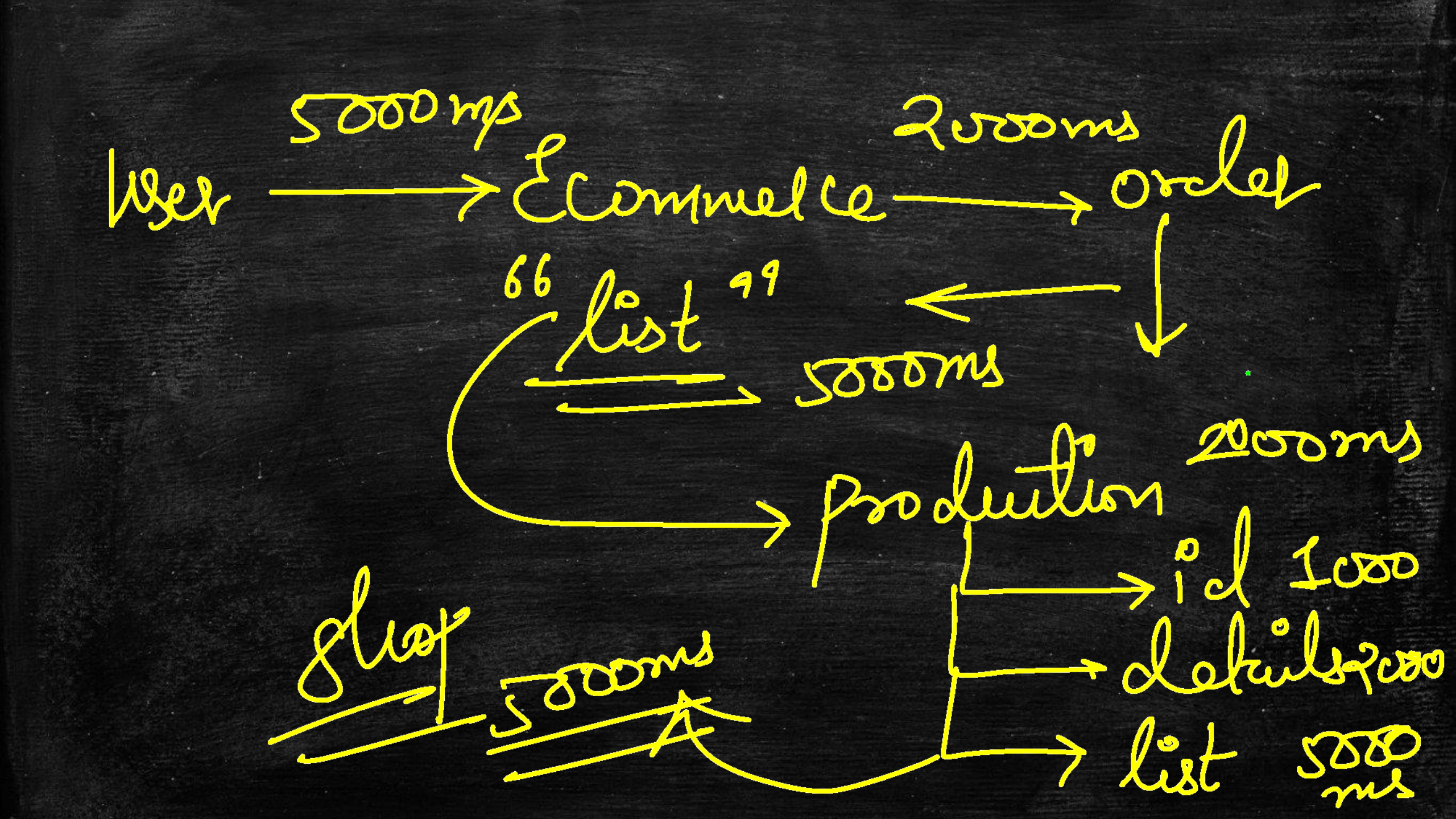
"Delay"

```
setTimeout(() => {
    console.log("process started");
});

setTimeout(() => {});
```

```
console.log("process2")
}
setTimeout(()=>{},5000)
,1000)
,2000)
```


2000
User
order
3000
Production
id → 2000
Tubel 2 → 3000
Tubel 3 → 1000
Tubel 4 → 2000
Exit → 5000


5000 ms
User
Ecommerce
2000ms
order
"list"
5000ms
product
2000ms
id 1000
details 2000
list 5000 ms
sleep
5000ms

Promises

```
const ticket = new Promise(
    function(resolve, reject)
    {
        const isBooked = false;
        if( isBooked )
```

```
{ resolve( ——— );
}

else {
    reject ("" ,"");
}
});
```

Then Catch Finally

Pizza → dough → cheese