# SQL

## Question 1 - Answer

select

users.first\_name

, users.last\_name

, count(\*)

from

user\_items items

left join users on users.id = items.user\_id

group by user\_id

;

## Question 2 - Answer

select \*

from users

where id not in (

select distinct(user\_id) from user\_items

)

;

## Question 3 - Answer

select

counts.description

, counts.num\_purchased

, counts.num\_purchased \* counts.cost\_pennies

from (

select

count(\*) as num\_purchased

, description

, cost\_pennies

from user\_items

join items on user\_items.item\_id = items.id

group by description

) as counts

;

## Question 4 - Answer

select

first\_name

, last\_name

, items.description

, user\_items.purchased\_at from user\_items

inner join ( select MAX(purchased\_at) purchased\_at, user\_id from user\_items group by user\_id) A on A.user\_id = user\_items.user\_id AND A.purchased\_at = user\_items.purchased\_at

join users on user\_items.user\_id = users.id

join items on user\_items.item\_id = items.id

order by user\_items.purchased\_at desc

## Question 5 – Answer

First I’d ask some more questions, if it happens to everyone from Customer Service team or just some members. If the issue appeared recently. I’d try to figure out if it is DB problem, or application problem. I’d check which queries are used and then test them in DB client. If there is delay and problem appeared recently then I’d check code repo if there were any changes to query. In this case query optimization might work. Checking DB’s issues might reveal that it is DB bug and DB server would need to be updated. There also might be option to add index on last name column.

If query gets results fast, then I’d go on to check application part. I’d try to call the API and check how long it takes. If it takes long, then it depends on the infrastructure and if data are passed through more applications.

If there is any middleware that’s logging request and response times, I’d check times of call there.

In worst case scenario it would be possible to split DB tables to contain user data only for specific first letter of the last name. But since that would require data migration, application change and database change I’d consider this as a last option if no other solution would be available.

## Question 6 – Answer

To be honest I have no idea. I’d google it and consult senior database expert. Might be problem with file system, or data corruption? I’d look at the data at some other time range that don’t have this problem and then check what’s different.

## Question 7 – Answer

I believe that is what transaction are for, either it all succeeds or fails and then rolls back. So I’d do the update without commit, then insert. If both operations were successful then I’d commit changes. If 2nd step fails, roll back. If 1st step fails, then application should report error to front end so error can be displayed to client.

# SCRIPTING

## Question 1 - Answer

cat data.dat | cut --delimiter=' ' -f3,8 --output-delimiter=' -> '

## Question 2 - Answer

cat data.dat | cut --delimiter=' ' -f5 | sort | uniq -c | sort -r | awk '{x = $1; $1 = $2; $2 = ","; $3 = x; print; }' | tr -d ' '

# Programming

## Question 1 - Answer

https://github.com/vildapavlicek/DiceRoller

## Question 2 - Answer

For1 mil requests I’d probably use buffered channel which I’d fill with requests with for loop and use N go routines to make requests, parse data and then store them to another channel from which another N go routines would take results and store them to DB. Doing counts and sorting values would probably be faster using database query than doing it in memory with arrays. Also setting transport so connections are reused instead of always creating one. I’d also expect site to have some graylisting or blacklisting mechanism as protection so it might need more tweaking so all calls are processed. Some func for requeuing if GET fails would be good idea.

# BUG Hunting

# Question 1 - Answer

var allMax uint = 0 and var max uint = 0 should be var x uint, all values are initialized to their default values, for int it’s 0, string it’s “” and bool is false.

Go func() uses “i” but doesn’t require it as input and it is not passed.

All prints miss “fmt.”

At imports statement, there is “.” (dot) in front of “fmt”

parallelism, \_ := strconv.Atoi(os.Args[1]) I think that error shouldn’t be ignored as it crashes program when os.Args is not number, I think it would be better to decide on default value to use conversion fails

fmt.Printf("error finding max for slice %d to %d, skipping this slice: %s\n", err) – not enough arguments, formatting requires digit, digit, string but has only err

correct way to split the array for go routines:

endIndex := perGoRoutine\*i + perGoRoutine

if i+1 == parallelism {

endIndex = len(numbers)

}

I think it would be probably better to ensure that argument for N go routines isn’t bigger than length of the array

For sorting it might be better to use int instead of uint and sort.Ints instead of range in range

Move mutex.Lock below the the err checking if

## Question 2 – Answer

I’d wrap all variables set in main in struct and do methods with pointer receiver.

Instead of using anonymous function as go routine it might be better to name it and make normal function from it then just call it with go argument

Set default value if strconv.Atoi fails and don’t ignore error

I’d probably make producing loop feeding slice parts to chan where N go routines would consume them and process.

Also write tests with examples might help (especially if you generate documentation)