

1. Feladat

Gábor beillesztésével kezdtem, mivel Péter és István is adva volt.

Gábor ezek szerint H, K, SZ, CS, - , SZO, V tud dolgozni.

Mivel Gábor péntekenként nem tud dolgozni, de Tamás igen, ebből következik, hogy szegény Richárdnak minden nap lehetősége van dolgozni.

Egy cégnél a munkavállalók az alábbi táblázat szerinti beosztásban dolgoznak:

Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat	Vasárnap
Péter <i>G</i>	István <i>G</i>	Péter <i>G</i>	István <i>G</i>	Tamás	István <i>G</i>	István <i>G</i>
Tamás <i>R</i>	Péter <i>R</i>	Tamás <i>R</i>	Péter <i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>

Richárd bármelyik nap tud dolgozni, amikor Tamás vagy Gábor dolgozik.

Gábor bármelyik nap tud dolgozni, amikor Péter vagy István dolgozik.

Melyikük tud a hét minden (bármely) napján dolgozni?

2. feladat:

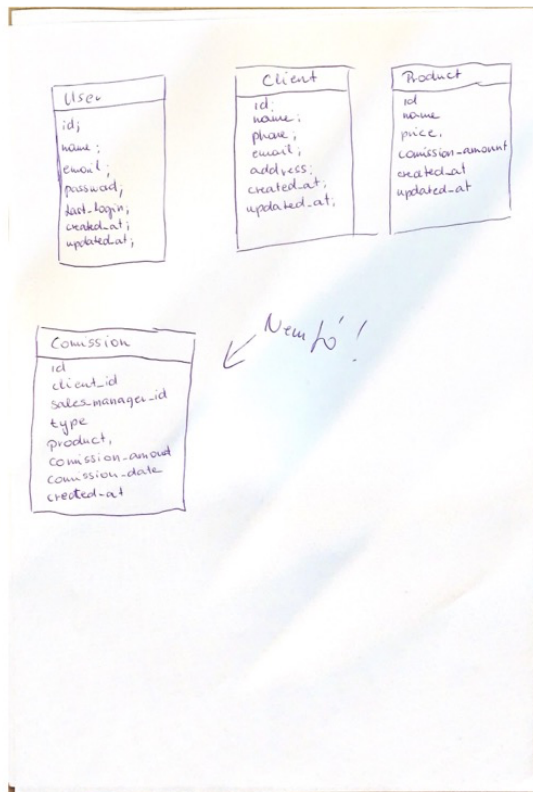
Egy folyamatban 6 lépés követ egymást, amelyeket A-F jelöljük.

F	B ?	A	? D	C	? F
---	-----	---	-----	---	-----

Az alábbi állításokat tudjuk:

- Az 'A' és 'D' lépések egymást követik (valamilyen sorrendben) *(2)*
- A 'B' lépés megelőzi az 'A' lépést *(1)*
- Az 'F' lépés nem szomszédos a 'C' lépéssel

Melyik lépés van a '?' helyén? *F*



Nem használnék `client_id`-t és `sales_manager_id`-t. Ehelyett olyan recordokat vennék fel amivel akár 5 év múlva is tudom, hogy mi volt ez a tranzakció hiszen, ha töröljük az ügyfeleket vagy a sales managereket, akkor nem sokra megyünk az `id`-val. Illetve a `commission_date` és a `created_at` recordot sem értem, hogy minek kettő.

DE

Felvetül egy kérdés a GDPR miatt. Amennyiben az ügyfél kéri az adatai törlését, akkor viszont az sem jó, mert törölni kell a `commission` táblából is. Ezt mindenképp egyeztetni kell egy ügyvéddel és az alapján meghozni a megfelelő döntést.

6. feladat:

A “*Commission*” entitásnak miért nincs `updated_at` mezője?

Mivel stornozni is lehet, ezért ezt az infót is külön kell tárolni, hogy minden pénzügyi tranzakció listázható legyen.

7. feladat:

Igen sérti. Fentebb leírtam, hogy ezzel mi lehet a gond.

8. feladat:

Itt bevallom volt gond. Lennének kérdéseim

Legtöbb eladott termék összesen :

```

SELECT product, sales_manager_id, COUNT(id) AS total_sales, SUM(commission_amount) as
total_amount FROM commission WHERE type != 'storno' AND `commission_date` BETWEEN
'2022-03-11' AND '2022-05-11' GROUP BY product, sales_manager_id ORDER BY
total_sales DESC LIMIT 3
  
```

Termékenként :

```
SELECT product, COUNT(id) AS total_sales, SUM(commission_amount) as total_amount  
FROM `comission` WHERE type != 'storno' AND commission_date BETWEEN '2022-03-11'  
AND '2022-05-11' GROUP BY product ORDER BY total_sales DESC LIMIT 3
```

1. akik mindegyikből legalább 10-et eladtak

összesen

```
"SELECT product, sales_manager_id, COUNT(id) AS total_sales, SUM(commission_amount)  
as total_amount FROM `comission` WHERE type != 'storno' AND commission_date BETWEEN  
'2022-03-11' AND '2022-05-11' GROUP BY product,sales_manager_id HAVING COUNT(id) >=  
10 ORDER BY total_sales DESC LIMIT 3";
```

Termékenként

```
SELECT product, COUNT(id) AS total_sales, SUM(commission_amount) as total_amount  
FROM `comission` WHERE type != 'storno' AND commission_date BETWEEN '2022-03-11'  
AND '2022-05-11' GROUP BY product HAVING COUNT(id) >= 10 ORDER BY total_sales DESC  
LIMIT 3
```

- összes eladott termék értéke alapján (bár itt azért kétségeket vet fel, hogy név alapján keressük az egyezést.

```
"SELECT comission.product, product.price, comission.sales_manager_id,  
COUNT(comission.id) AS total_sales, SUM(comission.commission_amount) as  
total_amount FROM `comission` INNER JOIN product ON (product.name =  
comission.product) WHERE comission.type != 'storno' AND comission.commission_date  
BETWEEN '2022-03-11' AND '2022-05-11' GROUP BY  
comission.product,comission.sales_manager_id,product.price HAVING  
COUNT(comission.id) >= 1 ORDER BY product.price DESC LIMIT 3";
```

```

$percent = 5/100;
$top_sales = Comission::getTopSalesBySalesManager();

$premium_sales = [];
foreach ($top_sales as $sale)
{
    // Calculate premium commission amount
    $premium_amount = $sale->commission_amount * $percent;

    if (isset($premium_sales[$sale->sales_manager_id]['premium'])){
        $premium_sales[$sale->sales_manager_id]['premium'] += $premium_amount;
    }else{
        $premium_sales[$sale->sales_manager_id]['premium'] = $premium_amount;
    }
}

// Add premium commission data to Database
foreach ($premium_sales as $sales_manager_id => $premium_amount)
{
    $premiumData = [
        'client_id' => 0,
        'sales_manager_id' => $sales_manager_id,
        'type' => 'premium',
        'product' => null,
        'commission_amount' => $premium_amount,
        'commission_date' => date('Y-m-d H:i:s'),
        'created_at' => date('Y-m-d H:i:s')
    ];

    Comission::addComission($premiumData);
}

```