

Praktiskais uzdevums vakances "Programmētājs" kandidātiem.

Uzdevums sastāv no divām daļām – .NET uzdevuma un SQL vaicājuma sagatavošanas. Uzdevuma mērķis ir gūt priekšstatu par kandidāta praktiskajām zināšanām. Tā izpildei nevajadzētu aizņemt vairāk kā 2h. Gadījumā, ja atvēlētajā laikā nav iespējams uzdevumu veikt pilnībā, var iesūtīt nepabeigtu risinājumu.

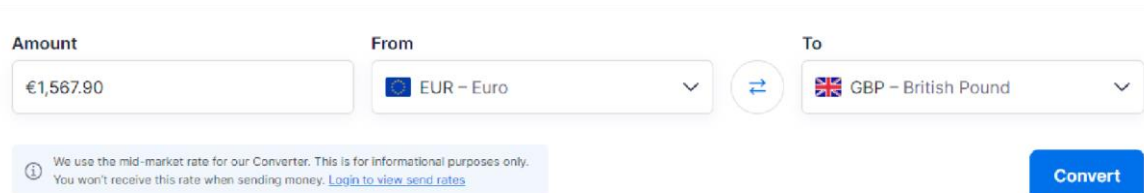
.NET uzdevumā prasīsim izveidot vienkāršotu valūtas maiņas kalkulatoru, kurā lietotājs izvēloties valūtas un konvertācijas summu, saņemtu konvertācijas rezultātu un aprēķinam izmantoto valūtas kursu. SQL uzdevumā prasīsim izveidot vienkāršus datu atlasē vaicājumus no teorētiskas darbinieku uzskaites datu bāzes.

.NET uzdevums

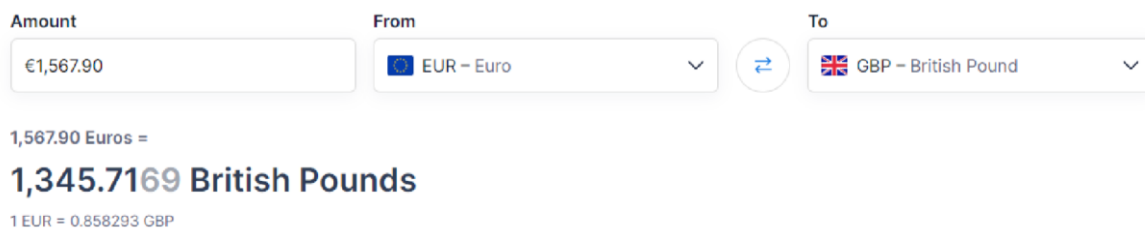
Izveidot vienkāršu WEB bāzētu valūtas maiņas kalkulatoru, kas izmanto publiski pieejamu REST API valūtas kursu iegūšanai vai konvertācijas rezultāta aprēķināšanai. Risinājums jāveido, izmantojot .NET Framework vai .NET Core (pēc brīvas izvēles). Risinājumā jāparāda prasmes izmanto OOP (piemēram, izveidojot klasi valūta, kas glabātu valūtas apzīmējumu, nosaukuma tekstuālo vērtību utt.). Lielāks uzsvars liekams uz backendu, frontends var būt ļoti vienkāršs (klienta puse izstrādājama html/ css/ JS (vanilla, jQuery vai kāds no JS ietvariem pēc brīvas izvēles)).

Funkcionālās prasības:

1. Valūtas maiņas kalkulators jāizveido izmantojot jebkuru REST API, kas ļauj iegūt valūtu kursus vai uzreiz aprēķināt konvertācijas rezultātus. Ieteicams izmantot šo risinājumu:
API: <https://www.exchangerate-api.com/>
API Key: your_secret_api_key
2. Lietotāja interfeisā jāizveido 3 ievdes lauki – konvertācijas valūta no (dropdown), konvertācijas valūta uz (dropdown), konvertējamā summa (teksta lauks, kurā atļauts ievadīt tikai ciparus un decimālo atdalītāju).
3. Kalkulācijai jāatbalsta vismaz 5 dažādas valūtas – EUR, USD, GBP un vismaz vēl 2 pēc brīvas izvēles. Valūtu izvēle ir atkarīga no izvēlētajā API pieejamajām valūtām (nevajadzētu valūtu izvēles laukos izmantot valūtas, kas API nav atbalstītas).



4. Attēlojot aprēķina rezultātu jānorāda arī izmantotais valūtas kurss.



Nefunkcionālās prasības:

1. Jāizstrādā testa piemērs, kurā novērtētu, vai konvertācijas rezultāts ir patiess.
2. Konvertācijas rezultāts jāaprēķina un API izsaukums jāveic backendā.

SQL uzdevums

Dotas tabulas:

#tmpPersonas – darbinieki, kas pieņemti darbā (unikālais identifikators, vārds, uzvārds)

#tmpNodalas – uzņēmumā darbojošās nodaļas (identifikators, nosaukums)

#tmpStrada – informācija, kurā nodaļā strādā konkrētais darbinieks (nodaļas identifikators, personas identifikators, datums, kad persona uzsāka strādāt konkrētajā nodaļā, datums, kad persona pārtrauca strādāt). Vienkāršības labad pieņemsim, ka darbinieks strādā tikai vienā nodaļā un darba attiecību laikā nodaļas, kurā tas strādā, nemainās.

Zemāk doti vaicājumi uzdevuma izpildīšanai – tabulu izveidošana, aizpildīšana ar datiem.

Izveidot vaicājumus, kas:

1. Atlasa visus darbiniekus, kas nostrādājuši vismaz 2 pilnus gadus, ieskaitot tos darbiniekus, kas no darba aizgājuši (kolonna BeiguDatums ir aizpildīta).

Atlasot norādīt personas vārdu, uzvārdu, nodaļas nosaukumu un periodu, kurā darbinieks strādājis attiecīgajā nodaļā, formātā dd.MM.yyyy – dd.MM.yyyy (ja darbinieks strādā vēl šobrīd (nav aizpildīta kolonna BeiguDatums), tad beigu datuma vietā liek tekstu "(aktuāls)"). Dati jāsakārto augošā secībā pēc datuma, kad darbinieks sācis strādāt nodaļā

Sagaidāmais rezultāts:

	Vards	Uzvards	Nosaukums	Periods
1	Ilze	Kakliņa	Grāmatvedība	10.04.2002 - 01.01.2019
2	Pēteris	Liepiņš	Front desk	17.02.2003 - 03.04.2007
3	Jānis	Bērziņš	Saimniecības daļa	06.12.2005 - (aktuāls)
4	Inta	Brūvere	Personāldaļa	11.01.2007 - (aktuāls)
5	Krišs	Kalniņš	Personāldaļa	18.03.2007 - (aktuāls)
6	Anna	Liepa	Front desk	12.01.2015 - 01.02.2017
7	Žanis	Zemzars	Saimniecības daļa	01.05.2016 - (aktuāls)

2. Atlasīt darbinieku, kas nostrādājis visilgāko laiku, norādot to pašu informāciju, ko iepriekšējā punktā plus dienu skaitu, ko darbinieks nostrādājis.

Sagaidāmais rezultāts:

	Vards	Uzvards	Nosaukums	Periods	Dienas
1	Ilze	Kakliņa	Grāmatvedība	10.04.2002 - 01.01.2019	6110

```
begin
create table #tmpPersonas
(
    ID int,
    Vards nvarchar(250),
    Uzvards nvarchar(250)
);
```

```

create table #tmpNodalas
(
    ID int,
    Nosaukums nvarchar(250)

);
create table #tmpStrada
(
    Persona int,
    Nodala int,
    SakumaDatums datetime,
    BeiguDatums datetime
);
insert into #tmpPersonas (ID, Vards, Uzvars) values (1, 'Jānis', 'Bērziņš');
insert into #tmpPersonas (ID, Vards, Uzvars) values (2, 'Anna', 'Liepa');
insert into #tmpPersonas (ID, Vards, Uzvars) values (3, 'Krišs', 'Kalniņš');
insert into #tmpPersonas (ID, Vards, Uzvars) values (4, 'Tamāra', 'Andersons');
insert into #tmpPersonas (ID, Vards, Uzvars) values (5, 'Ilmārs', 'Pētersons');
insert into #tmpPersonas (ID, Vards, Uzvars) values (6, 'Jānis', 'Zirnis');
insert into #tmpPersonas (ID, Vards, Uzvars) values (7, 'Inta', 'Brūvere');
insert into #tmpPersonas (ID, Vards, Uzvars) values (8, 'Pēteris', 'Liepiņš');
insert into #tmpPersonas (ID, Vards, Uzvars) values (9, 'Ilze', 'Kakliņa');
insert into #tmpPersonas (ID, Vards, Uzvars) values (10, 'Žanis', 'Zemzars');

insert into #tmpNodalas (ID, Nosaukums) values (1, 'IT daļa');
insert into #tmpNodalas (ID, Nosaukums) values (2, 'Grāmatvedība');
insert into #tmpNodalas (ID, Nosaukums) values (3, 'Personāldaļa');
insert into #tmpNodalas (ID, Nosaukums) values (4, 'Saimniecības daļa');
insert into #tmpNodalas (ID, Nosaukums) values (5, 'Front desk');

insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (1, 4,
'2005-12-06', null);
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (2, 5,
'2015-01-12', '2017-02-01');
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (3, 3,
'2007-03-18', null);
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (4, 2,
'2020-07-14', '2021-01-01');
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (5, 1,
'2021-11-03', null);
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (6, 1,
'2019-12-09', null);
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (7, 3,
'2007-01-11', null);
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (8, 5,
'2003-02-17', '2007-04-03');
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (9, 2,
'2002-04-10', '2019-01-01');
insert into #tmpStrada (Persona, Nodala, SakumaDatums, BeiguDatums) values (10, 4,
'2016-05-01', null);

/*select * from #tmpPersonas select * from
#tmpNodalas select * from #tmpStrada*/

drop table #tmpPersonas;
drop table #tmpNodalas;
drop table #tmpStrada;

end

```