Import nově přijatých studentů do Evergreenu

Export nových studentů z ISu

Studijní agenda > hromadný výběr > Vybrat dotaz *„Výběr studentů podle roku přijetí (je nutné změnit rok)“*.

Dotaz je možné také zadat ručně:

```
[fak JABOK]
[stud_rok_prijeti 2018]
```

Po vypsání dotazu použít odkaz *„Zpět na Manipulace se seznamem vybraných studentů“*

Poté v sekci Studijní - výběr studentů a operací vybrat „Tisky“ > Seznamy, sestavy (návod k sestavám, Prezentátoru) >

Ve „Výběru z připravených popisů sestav“ vybrat popis „JABOK: Knihovna: import nově přijatých studentů“.

Popis lze vložit také ručně:

```
<SESTAVA>
<STUDENT><TFORMA>
<F_OSOBA_UCO/>,<F_OSOBA_JMENO F_JMENA="prijmeni"/>,<F_OSOBA_JMENO
F_JMENA="jmeno"/>,<F_OSOBA_EMAIL_PRIHLASKA/>,MAILING,<F_OSOBA_TRVBYD_ADRESA_
RADEK1/>,<F_OSOBA_TRVBYD_ADRESA_RADEK2/>,<F_OSOBA_TRVBYD_STAT/>,<F_OSOBA_TRV
BYD_ADRESA_PSC/>,<F_OSOBA_TRVBYD_TELEFON/>,111,3,<F_OSOBA_CISLO_OP/>,103,2,<
F_OSOBA_DAT_NAROZENI FORMAT="11"/>,false,2019-09-01,</TFORMA>
<DATA_OSOBA><OSOBA_STUDIUM STUDIUM="?"/></DATA_OSOBA>
</STUDENT>
</SESTAVA>
```

Jako formu výstupu zvolit „Textový soubor s kódováním UTF-8 (Unicode)

Import studentů do Evergreenu

Vytvoření tabulky s novými čtenáři

```
CREATE TABLE jabok_students (
    student_id integer,
    barcode text,
    last_name text,
    first_name text,
```

```
    email text,
    address_type text DEFAULT 'MAILING',
    street1 text,
    street2 text,
    city text,
    country text DEFAULT 'Česká republika',
    postal_code text,
    phone text,
    profile_id integer,
    ident_type integer,
    ident_value text,
    home_ou integer,
    net_access_level integer,
    date_of_birth date,
    active boolean,
    expire_date timestamp,
    stat_cat integer,
    stat_cat_entry integer);
```

Kopírování dat z CSV

```
COPY jabok_students (barcode, last_name, first_name, email, address_type,
street1, city, country, postal_code, phone, profile_id, ident_type,
ident_value, home_ou, net_access_level, date_of_birth, active, expire_date,
stat_cat, stat_cat_entry)
    FROM '/home/opensrf/Ctenari/data_pro_import.csv'
        WITH CSV HEADER;
```

Nalezení duplicit dle jména a data narození

```
SELECT actor.usr.first_given_name, actor.usr.family_name, actor.usr.dob,
jabok_students.first_name, jabok_students.last_name, jabok_students.barcode
FROM jabok_students INNER JOIN actor.usr ON jabok_students.date_of_birth IS
NOT DISTINCT FROM actor.usr.dob AND actor.usr.first_given_name IS NOT
DISTINCT FROM jabok_students.first_name;
```

Výstup s duplicitními hodnotami

```
 first_given_name | family_name |    dob     | first_name |  last_name  |
barcode
------------------+-------------+------------+------------+-------------+---
------
 Adam             | Novák       | 1991-11-30 | Adam       | Novák       |
1234
 Anna             | Nováčková   | 1996-05-07 | Anna       | Nováčková   |
```

```
2345
 Alice            | Nováková    | 1979-07-07 | Alice      | Nováková    |
3456
 Amos             | Nováček     | 1973-04-11 | Amos       | Nováček     |
4567
(4 rows)
```

Smazání všech vypsaných duplicitních čtenářů

```
DELETE  FROM jabok_students USING actor.usr WHERE
jabok_students.date_of_birth IS NOT DISTINCT FROM actor.usr.dob AND
actor.usr.first_given_name IS NOT DISTINCT FROM jabok_students.first_name;
```

Vložení do tabulky actor.usr

Pozn.: do tabulky musí být vloženo heslo

```
INSERT INTO actor.usr (
    profile, usrname, passwd, email, ident_type, ident_value,
first_given_name, family_name,
    day_phone, home_ou, claims_returned_count, net_access_level, dob,
expire_date, active)
    SELECT profile_id, barcode, email, email, ident_type, ident_value,
first_name,
    last_name, phone, home_ou, 0, net_access_level, date_of_birth,
expire_date, active FROM jabok_students;
```

Vložení úvodního barcode

(Totožné se student id)

```
INSERT INTO actor.card (usr, barcode)
    SELECT actor.usr.id, jabok_students.barcode
    FROM jabok_students
        INNER JOIN actor.usr ON jabok_students.barcode = actor.usr.usrname;

UPDATE actor.usr AS au SET card = ac.id FROM actor.card AS ac WHERE
au.id=ac.usr AND au.id between 2089 AND 2184;


INSERT INTO actor.usr_address (usr, street1, city, state, country,
post_code, address_type)
    SELECT actor.usr.id, jabok_students.street1, jabok_students.city,
jabok_students.country, jabok_students.country, jabok_students.postal_code,
jabok_students.address_type
    FROM jabok_students
    INNER JOIN actor.usr ON jabok_students.barcode = actor.usr.usrname;
```

```
UPDATE actor.usr SET expire_date = '2018-09-01 00:00:00+02' where id between 2089 AND 2184;
```

From:
https://eg-wiki.osvobozena-knihovna.cz/ - **Evergreen DokuWiki CZ**

Permanent link:
https://eg-wiki.osvobozena-knihovna.cz/doku.php/knihovna_jabok:import_novych_studenu?rev=1539092134

Last update: **2018/10/09 15:35**