Plansoft.org: Integracja z BAZUS

Autor Maciej Szymczak

wersja 2024.04

[Opis funkcjonalny 1](#_Toc162759802)

[Wprowadzenie danych w BAZUS 2](#_Toc162759803)

[Relacje pomiędzy grupami 3](#_Toc162759804)

[Pobranie danych do Plansoft.org 3](#_Toc162759805)

[Jak sprawdzić, czy przesyłanie danych działa? 4](#_Toc162759806)

[Planujemy zajęcia! 4](#_Toc162759807)

[Przesyłamy rozkład do BAZUS 4](#_Toc162759808)

[Raport: Co nie zostało wysłane do Bazusa? 5](#_Toc162759809)

[Dla informatyków: szczegóły techniczne 5](#_Toc162759810)

[Jak to zostało zrobione? 5](#_Toc162759811)

[Przesłanie danych słownikowych 6](#_Toc162759812)

[Przesłanie planu studiów 6](#_Toc162759813)

[Odesłanie ułożonego rozkładu zajęć 7](#_Toc162759814)

[Bazus\_sub\_map 7](#_Toc162759815)

[Zastosowane Oprogramowanie 8](#_Toc162759816)

[Instalacja połączenia 8](#_Toc162759817)

[Blokujemy możliwość ręcznego wprowadzania rekordów 11](#_Toc162759818)

[Pytania i odpowiedzi 12](#_Toc162759819)

[Co jest wysyłane do Bazus? 12](#_Toc162759820)

[Rozwiązywanie problemów 12](#_Toc162759821)

[Przeglądanie duplikatów 13](#_Toc162759822)

[Edytowanie Integration id 14](#_Toc162759823)

[Komunikat „Zdublowana kombinacja Przedmiot-typ” 17](#_Toc162759824)

[Po co jest parametr pCleanpMode w pakiecie Integration? 17](#_Toc162759825)

[ORA-00001: unique constraint 18](#_Toc162759826)

[Przedmioty zawierające znaki końca wiersza w nazwach 19](#_Toc162759827)

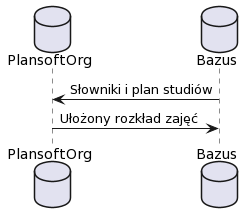
[Scalanie istniejącego przedmiotu z nowym przedmiotem 19](#_Toc162759828)

[Sprawdzenie, które pozycje z planu zajęć nie mogą być zaimportowane 20](#_Toc162759829)

# Opis funkcjonalny

Integracja opisana w tym dokumencie:

1. Przesyła dane słownikowe oraz plan studiów z BAZUS do Plansoft.org.
2. Następnie, po zaplanowaniu zajęć, przesyła gotowy rozkład zajęć do systemu BAZUS.



Ten rozdział opisuje czynności, jakie wykonujemy w kolejności chronologicznej.

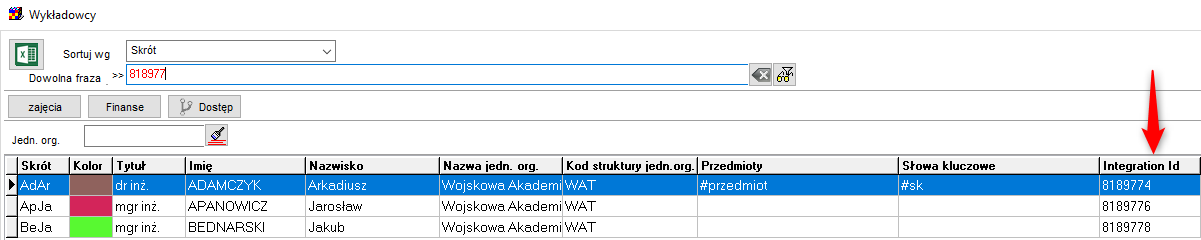
## Wprowadzenie danych w BAZUS

Następujące dane muszą zostać wprowadzone w systemie BAZUS:

* Słowniki: wykładowcy, grupy, sale, przedmioty, formy prowadzenia zajęć.
  + **Integration id**. Każdy rekord posiada tak zwany integration\_id, jest to unikatowy, stały identyfikator rekordu, który jest nadawany przez system BAZUS.

Integration id jest generowany automatycznie, nie musimy się o niego troszczyć.

Po przesłaniu danych ten numer możemy zobaczyć w Plansoft.org tutaj.



Możemy wyszukiwać rekordy za pomocą integration\_id wpisując nr w polu Dowolna fraza.

Jeżeli numer jest pusty to znaczy, że rekord nie był importowany z BAZUS, lecz został wprowadzony ręcznie. Mimo wszystko, takie rekord może być używany w planowaniu i zostanie on odesłany do Bazusa.

* + **Nieaktywne rekordy**.
    - Jeżeli rekord w BAZUS został zaznaczony jako nieaktywny, wówczas rekord staje nie nieaktywny również w Plansoft.org.
    - Jeżeli nieaktywny rekord w Plansoft.org został wcześniej zaimportowany z Bazus (to znaczy, nie jest to historyczny rekord), to dostęp do rekordu jest blokowany (rekord jest ukrywany).
    - Nieaktywny rekord nie jest aktualizowany (imię, nazwisko itd. nie są już aktualizowane)
  + Dane, których nie ma w Bazus (kolory, skróty) są generowane przez integrację automatycznie.
  + Importowane dane słownikowe są widoczne dla wszystkich planistów i ról.
* Plan studiów.
  + Do Plansoft.org przesyłane są tylko poprawne kombinacje, to znaczy takie, gdzie wszystkie pola są wypełnione: wykładowca, grupa, przedmiot, forma zajęć, liczba zajęć.

### Relacje pomiędzy grupami

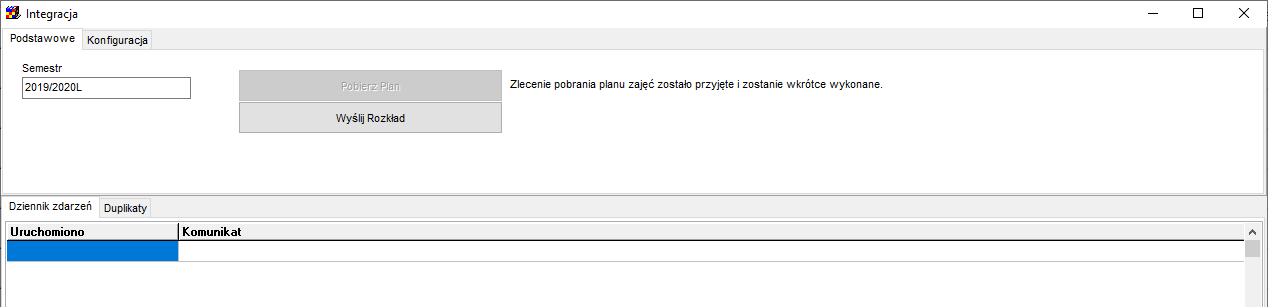
Relacje pomiędzy grupami (nadrzędny, podrzędny) nie są importowane z BAZUS.

Relacje należy utworzyć ręcznie w Plansoft.org. Relacje pomiędzy grupami zbudowane po stronie Plansoft.org nie są przenoszone do BAZUSa.

## Pobranie danych do Plansoft.org

W Plansoft.org musi utwórz semestr o nazwie odpowiadającej dokładnie nazwie semestru założonej w systemie Bazus. W razie, gdy semestr nie zostanie odnaleziony, informacja zostaje zaraportowana w logu.

Wprowadź aktualną nazwę semestru w polu Semestr na formularzu Plik | Integracja.



Synchronizacja danych słownikowych uruchamiana jest automatycznie godzinę- nic nie musisz robić.

Plan studiów aktualizowany jest również automatycznie, raz dziennie, o godz. 04.00.

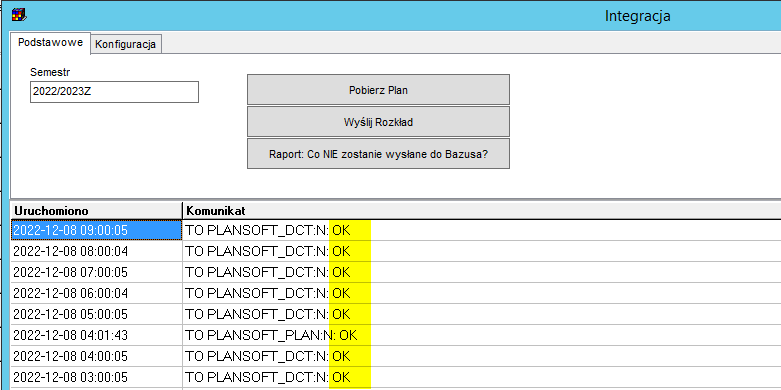
Jeśli chcesz pobrać plan szybciej, naciśnij przycisku **Pobierz Plan** na formularzu Plik | Integracja. W systemie rejestrowane jest zlecenie wysłania rozkładu, które zostanie zrealizowane o pełnej godzinie.

Import słowników trwa kilka sekund, import planu studiów trwa około dwóch minut i jest realizowany poza godzinami pracy z uwagi na konieczność przeliczenia liczby zaplanowanych zajęć dla całej bazy danych.

### Jak sprawdzić, czy przesyłanie danych działa?

W oknie **Plik | Integracja** sprawdzamy, czy były jakieś błędy w trakcie integracji oraz kiedy ostatni raz dane były przesyłane.

Nie musimy wczytywać się w znaczenie symboli na ekranie: ważne jest, tylko to, aby w kolumnie Uruchomiono była dzisiejsza data, a komunikat kończył się słowem „OK”.



## Planujemy zajęcia!

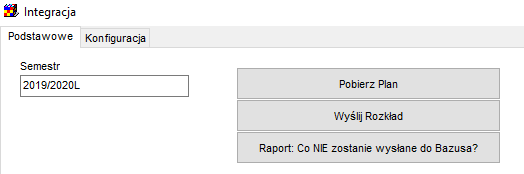
No cóż, wreszcie możemy rozpocząć planowanie ☺

## Przesyłamy rozkład do BAZUS

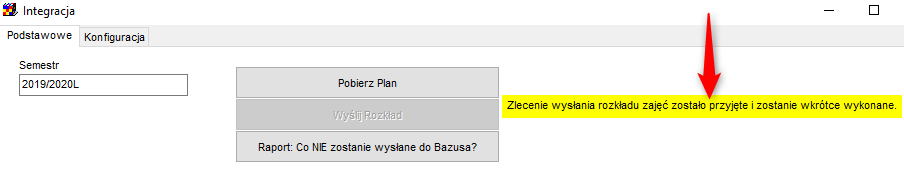
Rozkład przesyłamy za pomocą dobrze już znanego formularza **Plik | Integracja**.

Przycisk **Wyślij Rozkład** naciskamy, gdy rozkład jest ukończony.

Przycisk **Wyślij Rozkład** powinien być naciskany, gdy nikt inny nie pracuje w systemie Plansoft.org. Rozkład możemy przesyłać (aktualizować) wielokrotnie.



Aby wysłać rozkład naciskamy przycisk Wyślij Rozkład. Wysyłka realizowana jest o każdej pełnej godzinie.



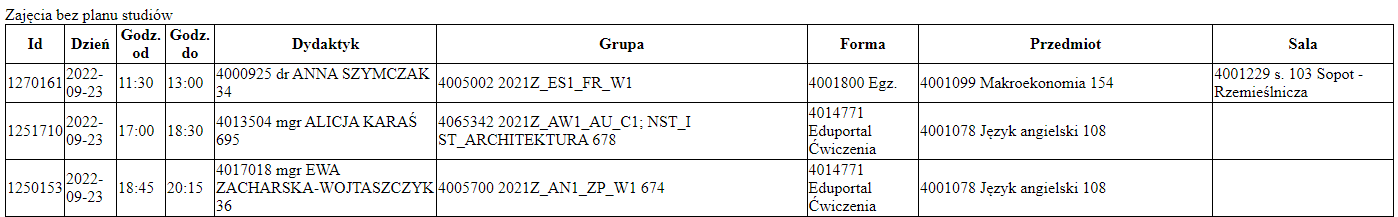
Przed wysłaniem, zajęcia są scalane (sąsiednie bloki są łączone).

### Raport: Co nie zostało wysłane do Bazusa?

Aby upewnić się, że wszystkie zajęcia zostały wysłane do Bazusa, naciskamy przycisk **Raport: Co nie zostanie wysłane do Bazusa?** Generowanie raportu trwa kilka minut.

Raport pokazuje zajęcia, których NIE można przesłać do Bazusa, gdy raport nie pokaże żadnych rekordów - to dobrze!

Przykładowy raport:



# Dla informatyków: szczegóły techniczne

|  |
| --- |
| W celu uruchomienia integracji z systemem Bazus konieczne jest uruchomienie funkcji integracyjnych w systemie Bazus, szczegóły należy uzgodnić z firmą Simple, dostawcą systemu Bazus. |

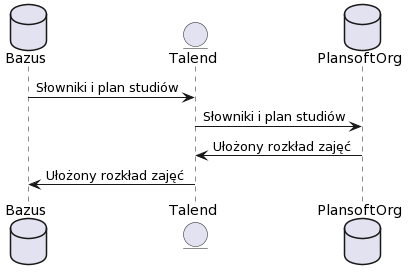
## Jak to zostało zrobione?

Integracja składa się z trzech części:

1. Przesłanie danych słownikowych,
2. Przesłanie planu studiów,
3. Odesłanie ułożonego rozkładu zajęć.

Program napisano w technologii Talend Data Integration Studio.

Dane są przesyłane pomiędzy bazami danych Oracle oraz MSSQL.



https://www.planttext.com/

|  |
| --- |
| @startuml  database Bazus  entity Talend  database PlansoftOrg  Bazus -> Talend : Słowniki i plan studiów  Talend -> PlansoftOrg : Słowniki i plan studiów  PlansoftOrg -> Talend : Ułożony rozkład zajęć  Talend -> Bazus : Ułożony rozkład zajęć  @enduml |

### Przesłanie danych słownikowych

**Kiedy**: Co godzinę.

|  |  |
| --- | --- |
| Przygotowanie danych w  MSSQL | Procedura osadzana:  IntPZ\_zestaw\_dydaktykow\_v1  IntPZ\_zestaw\_grup\_v1  IntPZ\_zestaw\_sal\_v1  IntPZ\_zestaw\_przedmiotow\_v1  IntPZ\_zestaw\_form\_zajec\_v1 |
| Przesłanie danych  MSSQL Odczyt -> Oracle Zapis | bazus.dbo.IntPZ\_zestaw\_dydaktykow -> int\_lecturers  bazus.dbo.IntPZ\_zestaw\_grup -> int\_groups  bazus.dbo.IntPZ\_zestaw\_sal -> int\_resources  bazus.dbo.IntPZ\_zestaw\_przedmiotow -> int\_subjects  bazus.dbo.IntPZ\_zestaw\_form\_zajec -> int\_forms |
| Przetwarzenie Oracle | Procedura osadzana:  begin integration.int\_to\_plansoft\_dict(); end; |

### Przesłanie planu studiów

**Kiedy**: Raz na dobę wieczorem oraz na żądanie użytkownika, o pełnych godzinach.

|  |  |
| --- | --- |
| Oracle: Sprawdzenie, czy interfejs powinien zostać uruchomiony. | select Value  from system\_parameters  where name = 'RUN\_INT\_TO\_PLANSOFT\_PLAN'  and value='YES' |
| Przetwarzanie MSSQL | Procedura osadzana:  IntPZ\_plan\_studiow\_v1 |
| Przesłanie danych MSSQL -> Oracle | bazus.dbo.IntPZ\_plan\_studiow->int\_plan |
| Przetwarzenie Oracle | Procedura osadzana:  begin integration.int\_to\_plansoft\_plan('N');  end; |
| Oracle  Potwierdzenie zakończenia przetwarzania | begin  delete from system\_parameters where name = 'RUN\_INT\_TO\_PLANSOFT\_PLAN';  commit;  end; |

### Odesłanie ułożonego rozkładu zajęć

**Kiedy**: Na żądanie użytkownika, o pełnych godzinach.

|  |  |
| --- | --- |
| Oracle: Sprawdzenie, czy interfejs powinien zostać uruchomiony. | begin  delete from system\_parameters where name = 'RUN\_INT\_TO\_PLANSOFT\_PLAN';  commit;  end |
| Przetwarzenie Oracle | begin integration.int\_from\_plansoft(); end;  Zajęcia są scalane (sąsiednie bloki są scalane) |
| Przesłanie danych Oracle -> MSSQL | int\_classes->**IntPZ\_plan\_zajec\_event**  int\_class\_members->**IntPZ\_plan\_zajec\_event\_member** |
| Przetwarzenie MSSQL | **Proedura osadzana**  **IntPZ\_planner\_to\_bazus** |
| Oracle  Potwierdzenie zakończenia przetwarzania | begin  delete from system\_parameters where name = 'RUN\_INT\_FROM\_PLANSOFT';  commit;  end |

### Bazus\_sub\_map

Bazus przekazuje przedmioty wymnożone przez formy prowadzenia zajęć, np. przedmiot matematyka przesyłany jest trzykrotnie, jeżeli prowadzonych jest w ramach trzech form zajęć.

Aby uniknąć wyświetlania wielokrotnie tego samego przedmiotu, do plansoft.org przedmiot jest importowany tylko jeden raz. W tym celu używamy tabeli mapującej bazus\_sub\_map: wiele przedmiotów w Bazus –jeden przedmiot w plansoft.org.

Najlepiej wyjaśnić mechanizm działania tabeli na przykładzie:

Bazus wysyła trzykrotnie ten sam przedmiot z trzema różnymi integration\_id:

|  |  |  |
| --- | --- | --- |
| Integration\_id | Przedmiot | Forma |
| 1 | Matematyka | Ćwiczenia |
| 2 | Matematyka | Wykład |
| 3 | Matematyka | Egzamin |

Do plansoft.org importujemy tylko jeden rekord (a nie trzy rekordy)

|  |  |
| --- | --- |
| Integration\_id | Przedmiot |
| 1 | Matematyka |

W tabeli bazus\_sub\_map tworzymy mapowanie:

|  |  |  |  |
| --- | --- | --- | --- |
| Integration\_id | Przedmiot | Forma | Plansoft. Przedmiot ID |
| 1 | Matematyka | Ćwiczenia | 1 |
| 2 | Matematyka | Wykład | 1 |
| 3 | Matematyka | Egzamin | 1 |

Mapowanie używane jest kilka razy:

1. Gdy zasilamy słownik Przedmioty
2. Gdy zasilamy słownik Plan Studiów
3. Gdy odsyłamy gotowy rozkład do Bazusa.

W plansoft.org planista planuje zajęcia:

|  |  |  |
| --- | --- | --- |
| Kiedy | Przedmiot | Forma |
| 2022.12.20 godz.8.00 | Matematyka (Id=1) | Wykład |
| 2022.12.20 godz.8.00 | Matematyka (Id=1) | Ćwiczenia |

Do bazusa wysyłany zajęcia:

|  |  |  |
| --- | --- | --- |
| Kiedy | Przedmiot | Forma |
| 2022.12.20 godz.8.00 | Matematyka (integration Id=2) | Wykład |
| 2022.12.20 godz.8.00 | Matematyka (integration Id=1) | Ćwiczenia |

### Zastosowane Oprogramowanie

Talend Data Integration Studio

Java AZUL

SQLDeveloper (ORACLE)

SSMS (Microsoft SQL Server Management Studio)

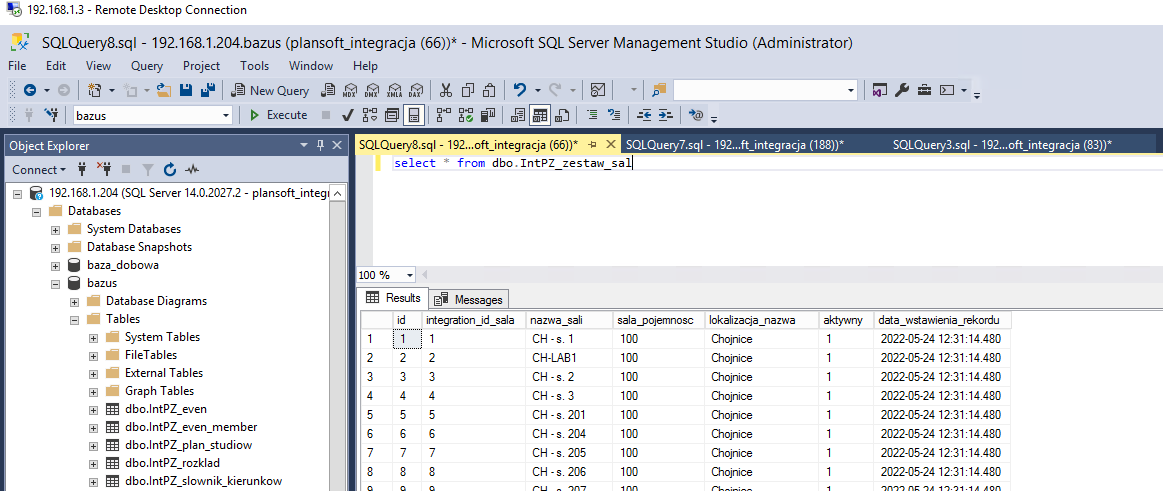
Notepad++

## Instalacja połączenia

1. Przed pierwszym uruchomieniem interfejsu uzupełniliśmy w plansoft.org integration\_id, miało to na celu uniknięcie utworzenia duplikatów rekordów po stronie plansoft.org. Zastosowano następujące reguły:
   1. Weryfikujemy, czy integration\_id generowany przez system źródłowy jest unikatowy.
   2. Porównujemy i uzupełniamy dane:

|  |
| --- |
| Wykladowcy istniejacy w Bazus i nie istniejacy w plansoft.org  select \* from  (  select title, first\_name, last\_name from int\_lecturers where is\_active=1  minus  select title, upper(first\_name), upper(last\_name) from lecturers  ) order by 3,2  Wykladowcy istniejacy w plansoft.org i nie istniejacy w Bazus  select \* from  (  select title, upper(first\_name), upper(last\_name) from lecturers where id >0  minus  select title, first\_name, last\_name from int\_lecturers where is\_active=1  ) order by 3,2  --------------------  -- LEC  update lecturers l set integration\_id = (select integration\_id from int\_lecturers where upper(first\_name)=upper(l.first\_name) and upper(last\_name)=upper(l.last\_name) and is\_active=1 );  update lecturers l set title = (select title from int\_lecturers where upper(first\_name)=upper(l.first\_name) and upper(last\_name)=upper(l.last\_name) and is\_active=1 )  where (select title from int\_lecturers where upper(first\_name)=upper(l.first\_name) and upper(last\_name)=upper(l.last\_name) and is\_active=1 ) is not null    --ROM  update rooms l set integration\_id = (select integration\_id from int\_resources where name=l.name and is\_active=1 ); --and location=l.attribs\_01  --GRO  update groups l set integration\_id = (select integration\_id from int\_groups where name=l.abbreviation ) and integration\_id is null;  --FOR  update forms set Name='(Nieaktywne)Praktyka', abbreviation='(X)Praktyka', is\_active='0' where Id=4007360;  update forms set name = initcap(name);  update forms m set integration\_id = (select integration\_id from int\_forms where name = m.name );  commit;  --GRO  update groups g set integration\_id = (select integration\_id from int\_groups where upper(name)=upper(l.first\_name) and upper(last\_name)=upper(l.last\_name) and is\_active=1);  --SUB  Ze względu na to, że Bazus zwraca kombinację: Przedmiot + forma, matchowanie nie było wykonywane |

1. Na serwerze integracyjnym instalujemy *Microsoft SQL Server Management Studio*, konfigurujemy połączenie i upewniamy się, ze możemy odczytać dane z tabel integracyjnych.



Upewniamy się, że poniższe zapytania SQL działają

|  |
| --- |
| **select** \* **from** bazus.dbo.IntPZ\_zestaw\_dydaktykow  **select** \* **from** bazus.dbo.IntPZ\_zestaw\_grup  **select** \* **from** bazus.dbo.IntPZ\_zestaw\_sal  **select** \* **from** bazus.dbo.IntPZ\_zestaw\_przedmiotow  **select** \* **from** bazus.dbo.IntPZ\_zestaw\_form\_zajec  **select** \* **from** bazus.dbo.IntPZ\_plan\_studiow |

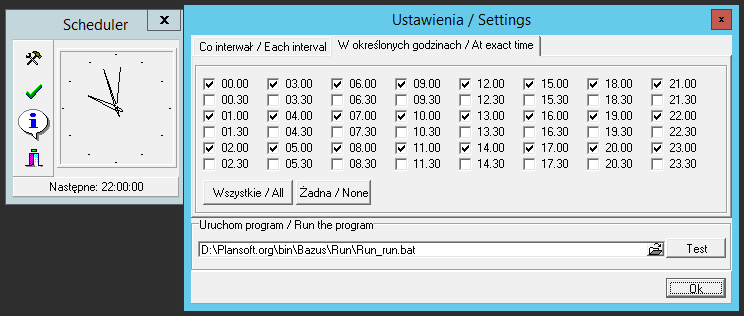
1. Aktywujemy integrację i konfigurujemy parametry integracji.

|  |
| --- |
| insert into system\_parameters (name, value) values ('INT\_IS\_ACTIVE', '1');  Commit; |

* Za pomogą formularza **Plik | Integracja** wprowadzamy Semestr i typ ograniczenia.

W polu **Typ ograniczenia** wybieramy: Semestr, Wykładowca, Przedmiot, Forma, Grupa.

1. Na serwerze harmonogramujemy usługę napisaną w Talend Data Integration Studio, która przesyła dane pomiędzy Bazusem (MSSQL) a Plansoft.org (Oracle).



D:\Plansoft.org\bin\Bazus\Run\Run\_run.bat

1. Harmonogramujemy proces synchronizacji planu zajęć.

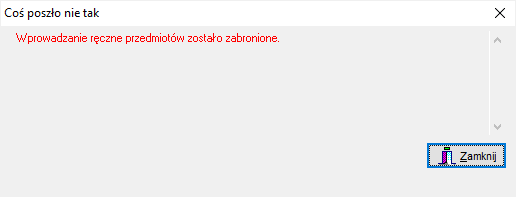
|  |
| --- |
| **Begin**  dbms\_scheduler.create\_job(  job\_name => 'INT\_BAZUS'  ,job\_type => 'PLSQL\_BLOCK'  ,job\_action => 'begin insert into system\_parameters (name, value) values(''RUN\_INT\_TO\_PLANSOFT\_PLAN'',''YES''); commit; end;'  ,repeat\_interval => 'freq=daily; byhour=3'  *--,repeat\_interval => 'freq=minutely'*  ,enabled => **TRUE**  ,comments => '');  *--DISPLAY SCHEDULED JOBS: select \* from dba\_scheduler\_jobs*  *--DROP JOB : begin dbms\_scheduler.drop\_job('INT\_BAZUS'); end;*  *--CLEAR LOG : delete from xxmsztools\_eventlog where module\_name = 'INT\_TO\_PLANSOFT';*  *--DISPLAY LOGS : select \* from xxmsztools\_eventlog where module\_name = 'INT\_TO\_PLANSOFT' order by id desc*  **end**; |

### Blokujemy możliwość ręcznego wprowadzania rekordów

Uruchom skrypt *Required\_integration\_id.sql*

Skrypt uruchamiamy dopiero po potwierdzeniu, że interfejs działa.

W efekcie, podczas próby dodania nowego przedmiotu użytkownik otrzyma następujący komunikat.



Podobny komunikat pojawi się podczas próby dodania grupy, wykładowcy, formy, sali.

## Pytania i odpowiedzi

### Co jest wysyłane do Bazus?

PYTANIE

================

W jaki sposób przesyłane jest do systemu Bazus zajęcie, które prowadzi kilku wykładowców, zajęcie jest dla kilku grup, w kilku salach, na przykład:

Data zajęcia: 2023.12.12 Godz. 9.00-10.00

Przedmiot: Matematyka

Rodzaj zajęć: Wykład

Wykładowcy: Janicki, Szymczak

Grupy: G1, G2

Sale: 115, 116

Czy przesyłane jest całe zajęcie? A może przesyłany jest tylko pierwszy wykładowca?

ODPOWIEDZ

================

Do systemu Bazus przesyłany jest kompletny opis zajęcia, zawierający wszystkie szczczegóły, w tym dwóch wykładowców, dwie grupy i dwie sale.

Data zajęcia, przedmiot, rodzaj zajęć i opis dla studentów zapisany jest w tabeli Bazus o nazwie "IntPZ\_plan\_zajec\_event"

Wykładowcy, grupy i sale - w sumie sześć rekordów, zapisane są w tabeli "IntPZ\_plan\_zajec\_event\_member"

Następnie uruchamiana jest procedura "IntPZ\_planner\_to\_bazus" po stronie systemu Bazus, która przenosi rekordy z tabel "IntPZ\_plan\_zajec\_event" i "IntPZ\_plan\_zajec\_event\_member" do docelowych tabel systemu Bazus.

## Rozwiązywanie problemów

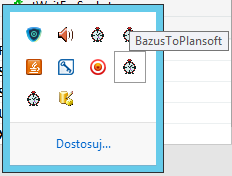
1. Za pomocą **Plansoft.org | Plik | Integracja** sprawdź, czy są nowe wpisy w logu.

Jeżeli podczas synchronizacji pojawia się błąd, to jest on zapisywany w logu synchronizacji.

Logi możesz również sprawdzić za pomocą SQL:

|  |
| --- |
| **select** \* **from** xxmsztools\_eventlog **where** module\_name = 'INT\_TO\_PLANSOFT' **order** **by** id **desc** |

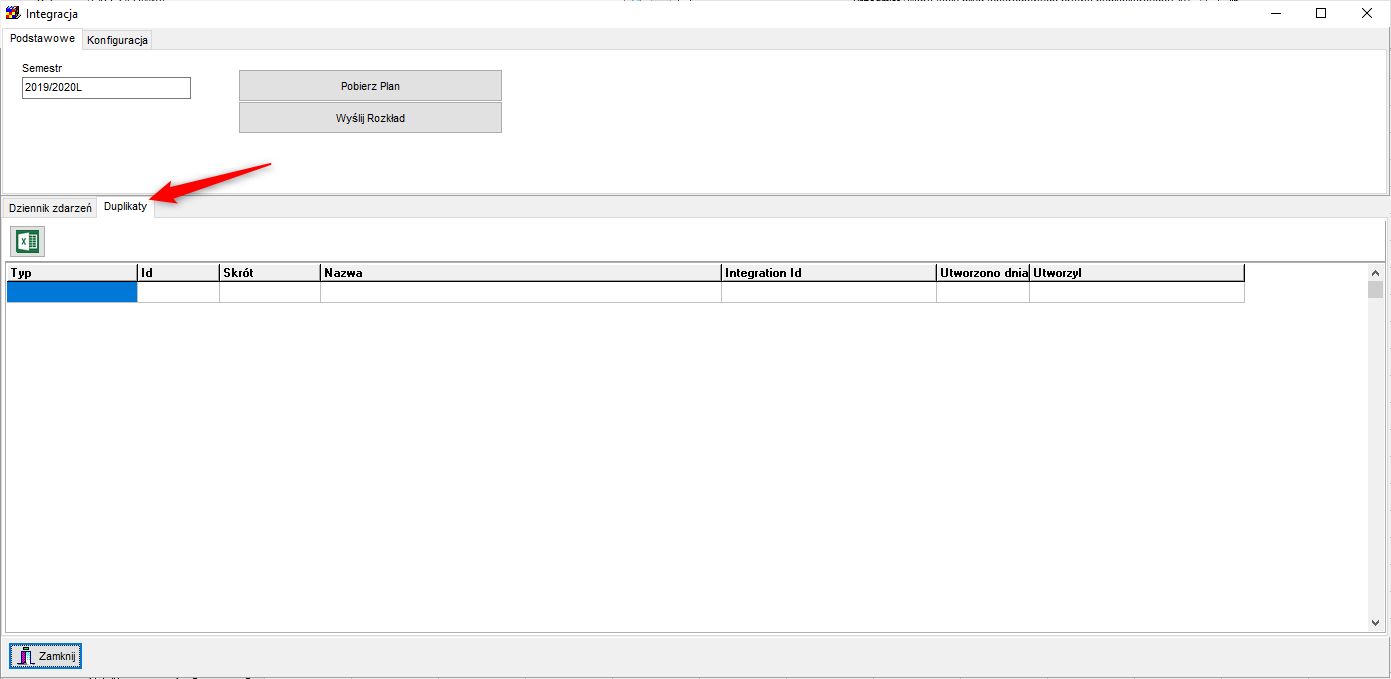
1. Zaloguj się do serwera 192.168.1.3. Sprawdź, czy zadanie jest uruchomione na serwerze. Jeżeli nie, to uruchom je



1. Zaloguj się do serwera 192.168.1.3. Uruchom *Talend Data Integration Studio*. Uruchom zadania recznie w celu przeźledzenia, czy wystepują błędy.

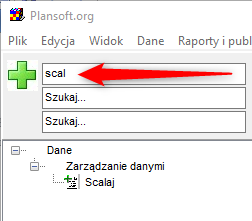
### Przeglądanie duplikatów

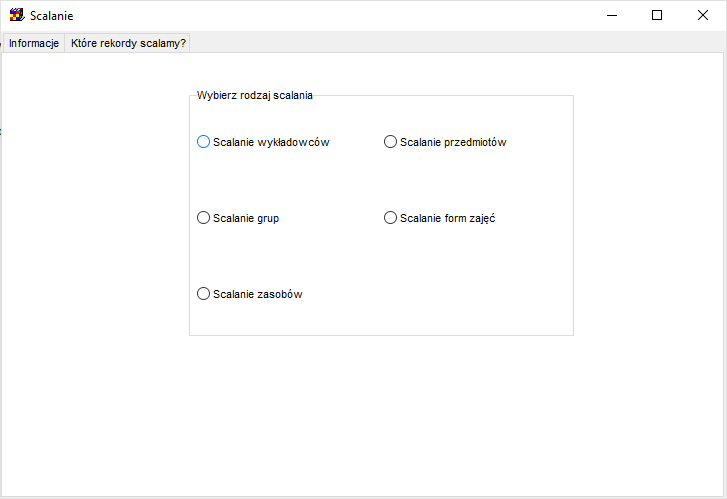
Aby sprawdzić, czy system Bazus utworzył duplikaty wykładowców, grup lub sal, uruchom okno Plik | Integracja, a następnie zaznacz zakładkę Duplikaty jak pokazano na rysunku poniżej.

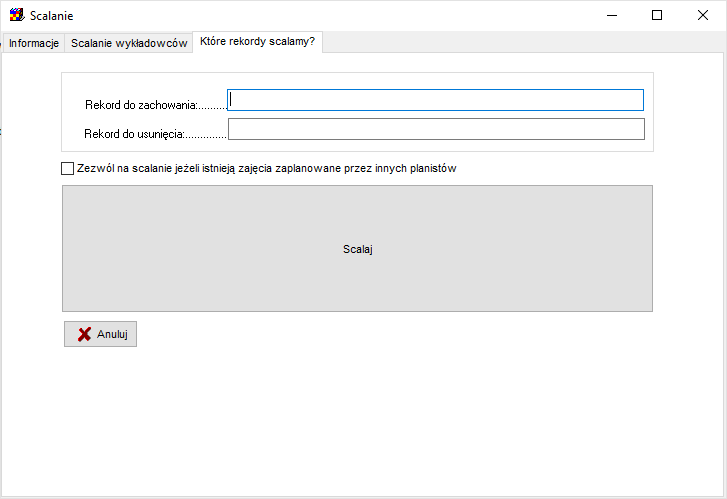


Jeżeli zakładka nie pokazuje żadnych rekordów, to znaczy, że nie ma duplikatów.

Jeżeli zakładka pokazuje duplikaty, to należy je jak najszybciej scalić za pomocą funkcji **Scalaj**.







Należy pozostawić rekord, który posiada wartość w polu **Integration Id** i usunąć rekord bez wartości w polu **Integration Id**. Jeżeli oba rekordy posiadają **Integration Id** to trzeba zdecydować o usunięciu któregoś z rekordów zarówno po stronie **Plansoft.org** jak i systemu **Bazus**.

Nieaktywne rekordy wykładowców nie są aktualizowane. Może to spowodować wystąpienie duplikatu, ze względu na przydzielenie skrótu innemu, aktywnemu rekordowi. W takim przypadku należy ręcznie zmienić skrót wykładowcy na nieaktywnym rekordzie.

### Edytowanie Integration id

*Integration ID* to wartość jednoznacznie identyfikująca rekord, jak numer PESEL człowieka.

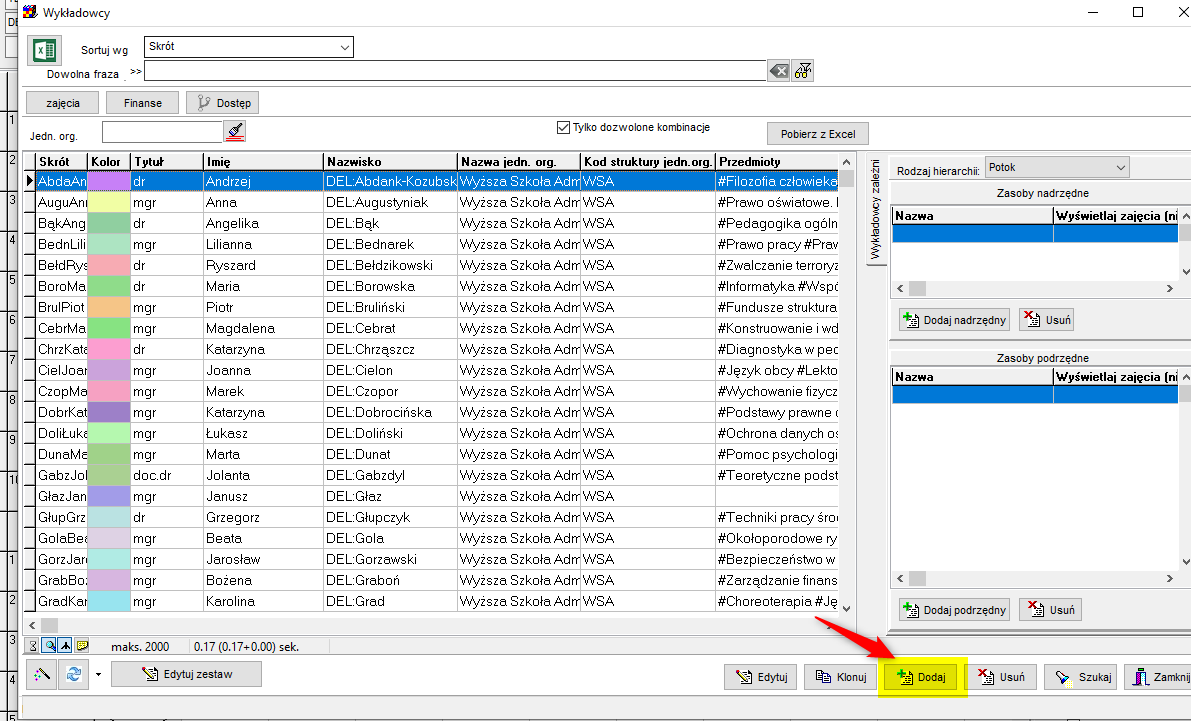
Jest to numer nadawany przez system, w którym powstał rekord, podobnie jak nr PESEL nadawany jest przez Urząd Stanu Cywilnego, gdy rodzi się nowy człowiek. Co do zasady otrzymujemy go raz i nie zmieniamy go.

Co do zasady, rekordy odpowiadający wykładowcom, grupom, salom, przedmiotom czy formom zajęć tworzone są w systemie zródłowym (Bazus / Usos / American Systems).

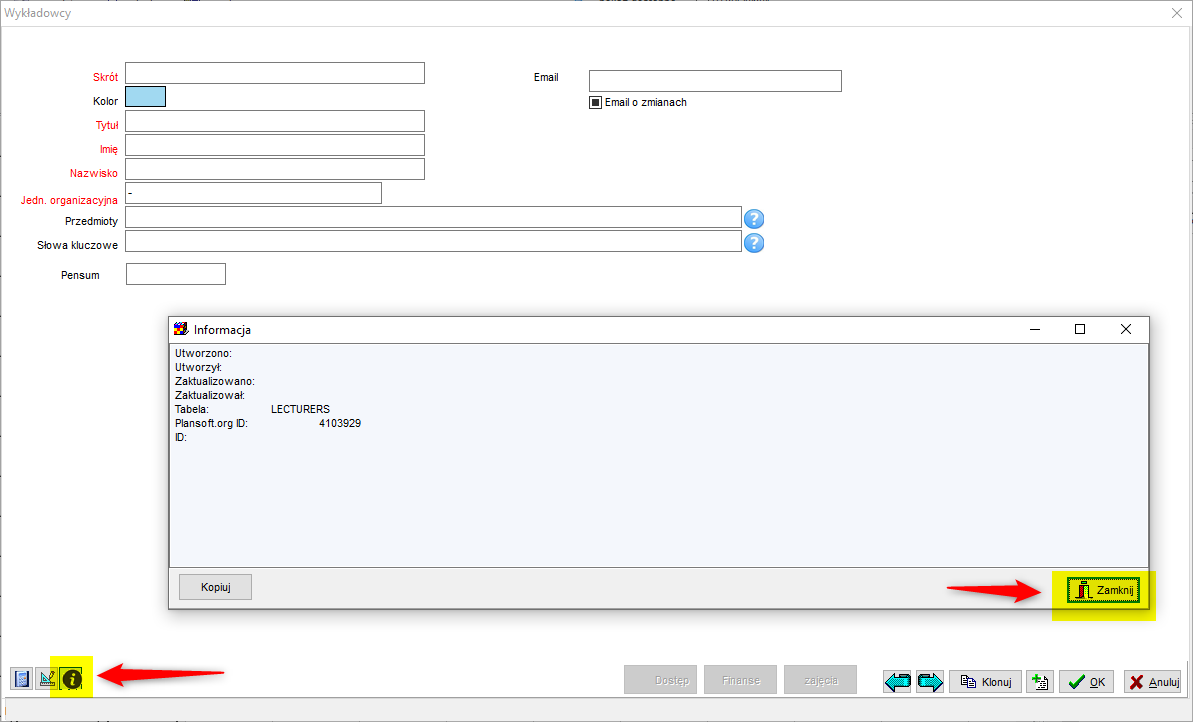
Co zrobić w sytuacji, kiedy musimy pilnie planować zajęcia, a z jakiegoś powodu system źródłowy nie przesłał utworzonego rekordu? Przede wszystkim powinniśmy zgłosić ten fakt obsłudze technicznej z prośbą o rozwiązanie problemu.

Jeżeli zależy nam jednak na czasie, brakujący rekord możemy wprowadzić w systemie Plansoft.org ręcznie, w następujący sposób:

1. Naciskamy przycisk **Dodaj** w formularzu **Dane | Wykładowcy**.



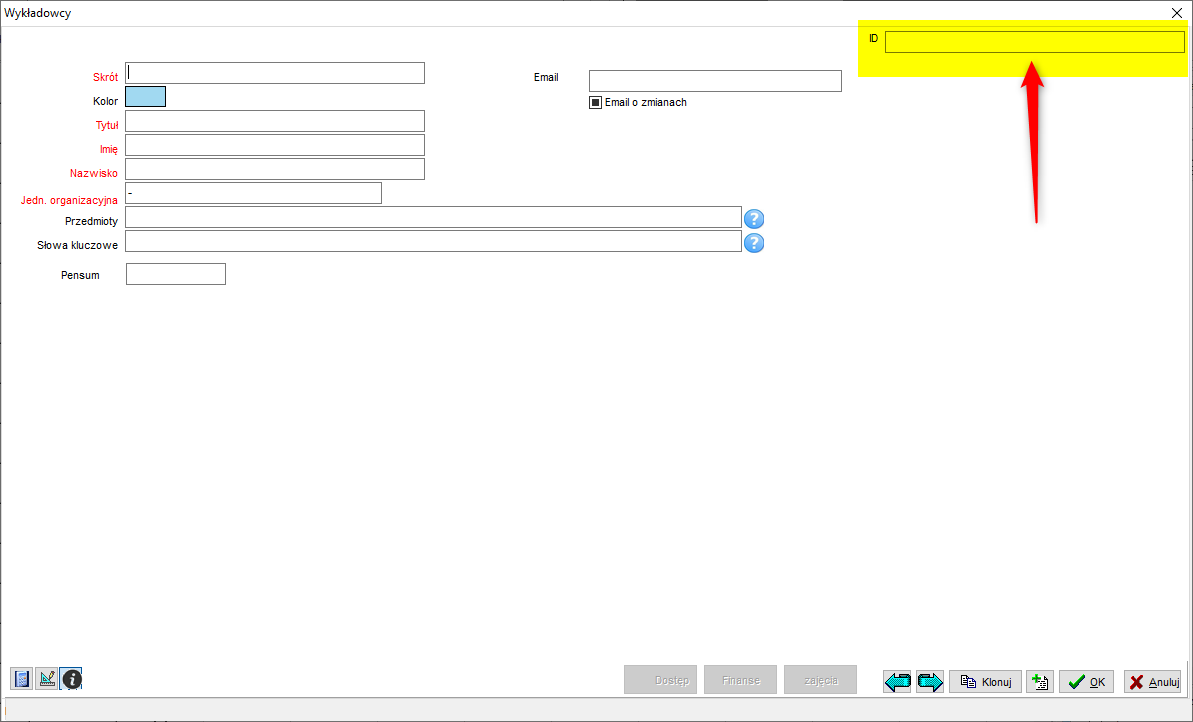
1. Naciskamy przycisk (I) i zamykamy okno Informacja, które pojawi się na ekranie.



1. Czynność ta spowoduje, że w górnym prawym rogu zobaczymy nowe pole ID. W polu tym wprowadzamy Integration ID. Zapisujemy rekord.

|  |
| --- |
| Ważne jest, żebyśmy wprowadzili Integration ID utworzony przez system źródłowy, nie może to być dowolna losowa wartość. Wpisanie dowolnej losowej wartości spowodowałoby utworzenie rekordu-duplikatu, gdy wreszcie system źródłowy prześle rekord. |

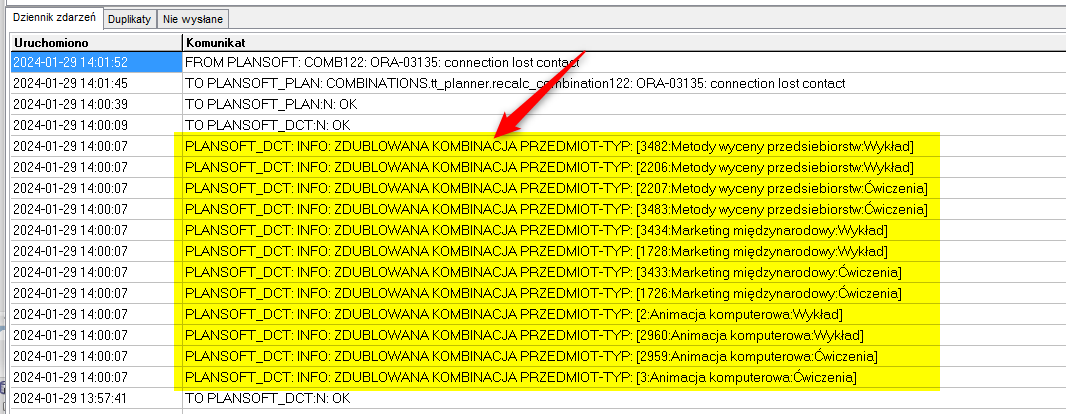
1. To wszystko, wykładowca został utworzony, można kontynuować pracę!



Opisana funkcjonalność działa również na formularzach Grupy, Sale, Przedmioty oraz formy zajęć.

### Komunikat „Zdublowana kombinacja Przedmiot-typ”

Komunikat oznacza, że w systemie Bazus omyłkowo przypisano przedmiotowi ten sam typ zajęć więcej niż jeden raz, np. „animacja komputerowa-ćwiczenia” i ponownie „animacja komputerowa-ćwiczenia”. Ten błąd nie powoduje zablokowania interfejsu, ale powinien zostać poprawiony w systemie bazus, ponieważ może powodować nieporozumienia na etapie raportowania danych.



### Po co jest parametr pCleanpMode w pakiecie Integration?

Parametr kasuje rekordy przed ich utworzeniem – w normalnym trybie pracy nie trzeba tego robić. Parametr może być pomocny podczas rozwiazywania problemów.

|  |  |
| --- | --- |
| Procedura | Akcja gdy pCleanpMode = true |
| **procedure** int\_to\_plansoft\_dict  (pCleanYpMode **varchar2** **default** 'N'); | **Truncate** **table** LEC\_PLA, GRO\_PLA, ROM\_PLA, SUB\_PLA **in** FOR\_PLA |
| **procedure** int\_to\_plansoft\_plan  (pCleanYpMode **varchar2** **default** 'N'); | **delete** **from** TT\_COMBINATIONS **where** integration\_id **is** **not** **null**  **delete** **from** tt\_resource\_lists **where** tt\_comb\_id **in** (**select** Id **from** TT\_COMBINATIONS **where** integration\_id **in** (**select** integration\_id **from** TT\_INTERFACE ) );  **delete** **from** TT\_INCLUSIONS **where** tt\_comb\_id **in** (**select** Id **from** TT\_COMBINATIONS **where** integration\_id **in** (**select** integration\_id **from** TT\_INTERFACE ) );  **delete** **from** tt\_cla **where** tt\_comb\_id **in** (**select** id **from** tt\_combinations **where** weight = 122 **and** per\_id = **nvl**(pPER\_ID, per\_id) **and** integration\_id **is** **not** **null**); |
| **procedure** int\_from\_plansoft  (pCleanYpMode **varchar2** **default** 'N'); | **delete** **from** tt\_cla **where** tt\_comb\_id **in** (**select** id **from** tt\_combinations **where** weight = 122 **and** per\_id = **nvl**(pPER\_ID, per\_id) **and** integration\_id **is** **not** **null**); |

### ORA-00001: unique constraint

LEC\_NAME\_UI, LEC\_ABBREVIATION\_I, SUB\_NAME\_UI, GRO\_ABBREVIATION\_I, SUB\_ABBREVIATION\_I, ROOM\_UK violated

|  |
| --- |
| *--Wyłącz blokadę anty-duplikatową*  **drop** **index** LEC\_NAME\_UI;  **drop** **index** LEC\_ABBREVIATION\_I;  **drop** **index** SUB\_NAME\_I;  **drop** **index** GRO\_ABBREVIATION\_I;  **drop** **index** SUB\_ABBREVIATION\_I;  **drop** **index** ROOM\_UK;  *--Uruchom integrację*  **begin** integration.int\_to\_plansoft\_dict(); **end**;  **select** \* **from** xxmsztools\_eventlog **where** module\_name = 'INT\_TO\_PLANSOFT' **order** **by** id **desc**  *--Przejrzyj błędy*  **select** \* **from** (  **select** 'LEC\_NAME\_UI' **type**, Id, ABBREVIATION, FIRST\_NAME||' '||LAST\_NAME||' '||TITLE **as** **name** , integration\_id, **to\_char**(creation\_date,'yyyy-mm-dd') creation\_date, created\_by  **from** LECTURERS **where** (FIRST\_NAME, LAST\_NAME, TITLE) **in** (**select** FIRST\_NAME, LAST\_NAME, TITLE **from** LECTURERS **group** **by** FIRST\_NAME, LAST\_NAME, TITLE **having** **count**(1)>1)  **union** **all**  **select** 'LEC\_ABBREVIATION\_I' **type**, Id, ABBREVIATION, FIRST\_NAME||' '||LAST\_NAME||' '||TITLE **as** **name**, integration\_id, **to\_char**(creation\_date,'yyyy-mm-dd') creation\_date, created\_by  **from** LECTURERS **where** (ABBREVIATION) **in** (**select** ABBREVIATION **from** LECTURERS **group** **by** ABBREVIATION **having** **count**(1)>1)  **union** **all**  **select** 'SUB\_NAME\_I' **type**, id, abbreviation, **name**, integration\_id, **to\_char**(creation\_date,'yyyy-mm-dd') creation\_date, created\_by  **from** subjects **where** **name** **in** (**select** **name** **from** subjects **group** **by** **name** **having** **count**(1)>1)  **union** **all**  **select** 'GRO\_ABBREVIATION\_I' **type**, Id, ABBREVIATION, **name** ||' ' || group\_type **as** **name**, integration\_id, **to\_char**(creation\_date,'yyyy-mm-dd') creation\_date, created\_by  **from** **groups** **where** ABBREVIATION **in** (**select** ABBREVIATION **from** **groups** **group** **by** ABBREVIATION **having** **count**(1)>1)  **union** **all**  **select** 'SUB\_ABBREVIATION\_I' **type**, id, ABBREVIATION, **name**, integration\_id, **to\_char**(creation\_date,'yyyy-mm-dd') creation\_date, created\_by  **from** subjects **where** (ABBREVIATION) **in** (**select** ABBREVIATION **from** subjects **group** **by** ABBREVIATION **having** **count**(1)>1)  **union** **all**  **select** 'ROOM\_UK' **type**, id, '' **as** ABBREVIATION, attribs\_01 || ' ' || **name** **as** **name**, integration\_id, **to\_char**(creation\_date,'yyyy-mm-dd') creation\_date, created\_by  **from** rooms **where** (**name**, attribs\_01) **in** (**select** **name**, ATTRIBS\_01 **from** rooms **group** **by** **name**, ATTRIBS\_01 **having** **count**(1)>1)  ) **order** **by** **type**, **name**, ABBREVIATION  *--Scal rekordy za pomoca standardowej funkcjonalnosci plansoft.org*  *--Bardziej skomplikowane przypadki:*  **begin**  *--reset abbr. I will be properly set by the interface*  **update** subjects **set** abbreviation='4325370' **where** id = 4325370;  **update** subjects **set** abbreviation='4046179' **where** id = 4046179;  **commit**;  **end**;  *--Ponownie aktywuj blokadę anty-duplikatową*  **CREATE** **UNIQUE** **INDEX** LEC\_NAME\_UI **ON** LECTURERS ("FIRST\_NAME", "LAST\_NAME", "TITLE") **TABLESPACE** "USERS" ;  **CREATE** **UNIQUE** **INDEX** LEC\_ABBREVIATION\_I **ON** LECTURERS ("ABBREVIATION") **TABLESPACE** "USERS" ;  **CREATE** **UNIQUE** **INDEX** SUB\_NAME\_I **ON** SUBJECTS (**NAME**) **TABLESPACE** "USERS";  **CREATE** **UNIQUE** **INDEX** GRO\_ABBREVIATION\_I **ON** **GROUPS** ("ABBREVIATION") **TABLESPACE** "USERS";  **CREATE** **UNIQUE** **INDEX** SUB\_ABBREVIATION\_I **ON** SUBJECTS ("ABBREVIATION") **TABLESPACE** "USERS";  **CREATE** **UNIQUE** **INDEX** ROOM\_UK **ON** ROOMS (**CASE** "RESCAT\_ID" **WHEN** 1 **THEN** "NAME"||' '||"ATTRIBS\_01" **ELSE** **TO\_CHAR**("ID") **END** ) **TABLESPACE** "USERS"; |

### Przedmioty zawierające znaki końca wiersza w nazwach

|  |
| --- |
| select id, name, integration\_id from subjects where ascii(substr(name,length(name),1))=10  update subjects set name = replace(name,chr(10),'') |

### Scalanie istniejącego przedmiotu z nowym przedmiotem

|  |
| --- |
| **update** subjects **set** **name** = **name** || '(X)' **where** integration\_id **is** **null**;  **select** to\_fix.\*  , 'update classes set sub\_id='||id\_to\_retain ||' where sub\_id='||it\_to\_delete||';' **sql**  **from**  (**select** Id it\_to\_delete  , **name**  , integration\_id  , (**select** id **from** subjects **where** **name** = m.**name**||'(X)' **and** integration\_id **is** **null** ) id\_to\_retain  **from** subjects m  **where** integration\_id **is** **not** **null** **and** **name** **in** (**select** **replace**(**name**,'(X)','') **from** subjects **where** integration\_id **is** **null** **and** **name** **like** '%(X)%')  ) to\_fix  **begin**  **for** rec **in** (  **select** to\_fix.\*  **from**  (**select** Id it\_to\_delete  , **name**  , integration\_id  , (**select** id **from** subjects **where** **name** = m.**name**||'(X)' **and** integration\_id **is** **null** ) id\_to\_retain  **from** subjects m  **where** integration\_id **is** **not** **null** **and** **name** **in** (**select** **replace**(**name**,'(X)','') **from** subjects **where** integration\_id **is** **null** **and** **name** **like** '%(X)%')  ) to\_fix  ) **loop**  **update** subjects **set** integration\_id = -rec.integration\_id **where** id=rec.it\_to\_delete;  **update** subjects **set** integration\_id = rec.integration\_id **where** id=rec.id\_to\_retain;  **delete** **from** subjects **where** id=rec.it\_to\_delete;  **commit**;  **end** **loop**;  **commit**;  **end**; |

### Sprawdzenie, które pozycje z planu zajęć nie mogą być zaimportowane

|  |
| --- |
| select int\_plan.\*  , (select count(id) from lecturers where integration\_id=integration\_id\_lec) lec\_ok  , (select count(id) from groups where integration\_id=integration\_id\_gro) gro\_ok  , (select count(id) from subjects where integration\_id=integration\_id\_sub) sub\_ok  , (select count(id) from forms where integration\_id=integration\_id\_for) for\_ok  from int\_plan where integration\_id in (  select integration\_id from int\_plan  minus  select integration\_id from tt\_combinations  )  and cycle\_name=**'2022/2023Z'**  and is\_active='1' |