Widoki

Course_information

Widok przedstawia informacje o stworzonych kursach dostępnych i nie dostępnych. Przedstawia również informacje o ilości modułów i maksymalnej ilości miejsc na kursie (wyznaczana na podstawie limitu w modułach stacjonarnych).

```
create view course_information as
    with count_modules as (select CourseID, count(ModuleID) as count_module
                           from Modules
                           group by CourseID
    ), limit_for_course as (select Courses.CourseID, min(Limit) as min_limit
                            from Courses inner join Modules on Courses.CourseID = Modules.CourseID
                            inner join In_person_Modules on Modules.ModuleID = In_person_Modules.ModuleID
                            group by Courses.CourseID)

    select Courses.CourseID,
           Name,
           Description,
           concat(FirstName, ' ', LastName) as Coordinator_full_name,
           StartDate,
           EndDate,
           isnull(count_module, 0) as Total_modules,
           min_limit as Total_places
    from Courses left join Employees on Courses.CoordinatorID = Employees.EmployeeID
    left join count_modules on Courses.CourseID = count_modules.CourseID
    left join limit_for_course on Courses.CourseID = limit_for_course.CourseID;
```

Employees_information

Widok przedstawia informacje danych pracowników w systemie.

```
CREATE view employees_information as
    select EmployeeID,
           FirstName,
           LastName,
           Phone,
           Email,
           Address,
           City,
           PostalCode,
           PositionName
    from Employees inner join Employees_Postions on Employees.PositionID = Employees_Postions.PositionID
```

Future_course_sign

Widok przestawia informacje o ilości osób zapisanych na przyszłe kursy. Przyszłe kursy oznaczają wydarzenia o dalszej dacie niż aktualna.

```
CREATE view future_course_sign as
    select future_courses.CourseID,
           Name,
           StartDate,
           EndDate,
           count(UserID) as Total_users
    from future_courses left join Users_Courses on Users_Courses.CourseID = future_courses.CourseID
    group by future_courses.CourseID, Name, StartDate, EndDate
```

Future_courses

Widok przedstawia informacje o przyszłych kursach.

```
create view future_courses as
    select *
    from course_information
    where getdate() < StartDate
```

Future_studie_sign

Widok przedstawia informacje o ilości osób zapisanych na przyszłe studia.

```
CREATE view future_studie_sign as
    select future_studies.StudiesID,
           Name,
           StartDate,
           EndDate,
           count(UserID) as Total_users
    from future_studies left join Users_Studies on Users_Studies.StudiesID = future_studies.StudiesID
    group by future_studies.StudiesID, Name, StartDate, EndDate;
```

Future_studies

Widok przedstawia informacje o przyszłych studiach.

```
create view future_studies as
    select *
    from studie_information
    where getdate() < StartDate
```

Future_webinar_sign

Widok przedstawia informacje o ilości zapisanych osób na przyszłe webinary.

```
CREATE view future_webinar_sign as
    select future_webinars.WebinarID,
           Name,
           DateAndBeginningTime,
           Duration,
           count(UserID) as Total_users
    from future_webinars left join Users_Webinars on Users_Webinars.WebinarID = future_webinars.WebinarID
    group by future_webinars.WebinarID, Name, Duration, DateAndBeginningTime
```

Future_webinars

Widok przedstawia informacje o przyszłych webinarach.

```
create view future_webinars as
    select *
    from webinar_information
    where getdate() < DateAndBeginningTime
```

In_person_module_information

Widok przedstawia informacje o modułach stacjonarnych w kursach.

```
create view in_person_module_information as
    select ModuleID,
           Classroom,
           concat(FirstName, ' ', LastName) as Translator_full_name,
           LanguageName,
           Limit
    from In_person_Modules left join Translators on In_person_Modules.TranslatorID = Translators.TranslatorID
    left join Languages on In_person_Modules.LanguageID = Languages.LanguageID;
```

Languages_count_translators

Widok przedstawia ilu tłumaczy mówi w danych językach.

```
create view languages_count_translators as
    select LanguageName,
           count(TranslatorID) as Translators_count
    from Languages
        inner join Translators_Languages on Languages.LanguageID = Translators_Languages.LanguageID
    group by LanguageName;
```

Module_information

Widok przedstawia informacje o modułach w kursach.

```
create view module_information as
    select ModuleID,
           concat(FirstName, ' ', LastName) as Teacher_full_name,
           CourseID,
           Name,
           Description,
           DateAndBeginningTime,
           Duration,
           TypeName
    from Modules inner join Employees on Modules.TeacherID = Employees.EmployeeID
    inner join Types on Modules.TypeID = Types.TypeID;
```

Online_sync_module_information

Widok przedstawia informacje o modułach online-synchronicznych w kursach.

```
create view online_sync_module_information as
    select ModuleID,
           MeetingLink,
           RecordingLink,
           concat(FirstName, ' ', LastName) as Translator_full_name,
           LanguageName
    from Online_Sync_Modules inner join Translators on Online_Sync_Modules.TranslatorID = Translators.TranslatorID
    inner join Languages on Online_Sync_Modules.LanguageID = Languages.LanguageID;
```

Online_async_module_information

Widok przedstawia informacje o modułach online-asynchronicznyhc w kursach.

```
create view online_async_module_information as
    select ModuleID,
           RecordingLink
    from Online_Async_Modules;
```

Studie_information

Widok przedstawia informacje o studiach wraz z ilością przedmiotów, ilością spotkań oraz maksymalną ilością miejsc na studiach (wyznaczaną po podstawie spotkań stacjonarnych).

```
create view studie_information as
    with count_subjects as (select StudiesID, count(SubjectID) as total_subjects
                            from Subjects
                            group by StudiesID
    ), count_meetings as (select StudiesID, count(MeetingID) as total_meetings
                          from Meetings
                                   inner join Subjects on Meetings.SubjectID = Subjects.SubjectID
                          group by StudiesID
    ), limit_for_studie as (select StudiesID, min(Limit) as min_limit
                            from In_person_Meetings inner join Meetings on In_person_Meetings.MeetingID = Meetings.MeetingID
                            inner join Subjects on Meetings.SubjectID = Subjects.SubjectID
                            group by StudiesID
    )

    select Studies.StudiesID,
           concat(FirstName, ' ', LastName) as Coordinator_full_name,
           Name,
           Description,
           StartDate,
           EndDate,
           Price,
           isnull(total_subjects, 0) as Total_subjects,
           isnull(total_meetings, 0) as Total_meetings,
           min_limit as Total_places
    from Studies left join Employees on Studies.CoordinatorID = Employees.EmployeeID
    left join count_subjects on Studies.StudiesID = count_subjects.StudiesID
    left join count_meetings on Studies.StudiesID = count_meetings.StudiesID
    left join limit_for_studie on Studies.StudiesID = limit_for_studie.StudiesID;
```

Translators_information

Widok przedstawia informacje o tłumaczach pracujących na platformie.

```
CREATE view translators_information as
    select TranslatorID,
           FirstName,
           LastName,
           Phone,
           Email,
           Address,
           City,
           PostalCode
    from Translators;
```

Translators_language

Widok przedstawia zestawienia tłumaczy wraz z językami, które tłumaczą

```
create view translators_language as
    select concat(FirstName, ' ', LastName) as fullName,
           LanguageName
    from Translators
        inner join Translators_Languages on Translators.TranslatorID = Translators_Languages.TranslatorID
        inner join Languages on Languages.LanguageID = Translators_Languages.LanguageID;
```

Users_information

Widok przedstawia informacje o użytkownikach systemu.

```
CREATE view users_information as
    select UserID,
           FirstName,
           LastName,
           Phone,
           Email,
           Address,
           City,
           PostalCode
    from Users;
```

Webinar_information

Widok przedstawia informacje o webinarach dostępnych czy nie dostępnych.

```
create view webinar_information as
    select WebinarID,
           Name,
           Description,
           DateAndBeginningTime,
           Duration,
           concat(C.FirstName, ' ', C.LastName) as full_name_coordinator,
           concat(T.FirstName, ' ', T.LastName) as full_name_teacher,
           concat(isnull(Translators.FirstName, ''), ' ', isnull(Translators.LastName, '')) as full_name_translator,
           Price,
           LanguageName,
           RecordingLink,
           MeetingLink

    from Webinars inner join Employees T on Webinars.TeacherID = T.EmployeeID
    left join Employees C on C.EmployeeID = Webinars.CoordinatorID
    left join Translators on Webinars.TranslatorID = Translators.TranslatorID
    left join Languages on Webinars.LanguageID = Languages.LanguageID;
```

Course_module_types

Widok ten przedstawia ile jest modułów każdego typu w każdym kursie.

```
create view course_module_types as
    with in_person_count as (
```

```
        select CourseID, count(Modules.ModuleID) as in_person
        from Modules inner join In_person_Modules on Modules.ModuleID = In_person_Modules.ModuleID
        group by CourseID
    ), online_sync_count as (
        select CourseID, count(Modules.ModuleID) as online_sync
        from Modules inner join Online_Sync_Modules on Modules.ModuleID = Online_Sync_Modules.ModuleID
        group by CourseID
    ), online_async_count as (
        select CourseID, count(Modules.ModuleID) as online_async
        from Modules inner join Online_Async_Modules on Modules.ModuleID = Online_Async_Modules.ModuleID
        group by CourseID
    )

    select Courses.CourseID, in_person, online_sync, online_async
    from Courses left join in_person_count on Courses.CourseID = in_person_count.CourseID
    left join online_async_count on Courses.CourseID = online_async_count.CourseID
    left join online_sync_count on Courses.CourseID = online_sync_count.CourseID
```

Course_passes

Widok ten przedstawia listę ze zdawalnością w modułach odpowiednich kursów dla osób zapisanych.

```
create view course_passes as
    select CourseID,
           Modules.ModuleID,
           FirstName,
           LastName,
           Passed
    from Users_Modules_Passes inner join Modules on Users_Modules_Passes.ModuleID = Modules.ModuleID
    inner join Users on Users_Modules_Passes.UserID = Users.UserID
```

Course_sing_limit

Widok przedstawia dla każdego kursu ile miejsc się na niego sprzedało i jaki jest limit.

```
create view course_sign_limit as
    with course_limits as (
        select CourseID, min(Limit) as total_limit
        from Modules inner join In_person_Modules on Modules.ModuleID = In_person_Modules.ModuleID
        group by CourseID
    )

    select Courses.CourseID, total_product_orders, total_limit
    from Courses left join products_orders on products_orders.ProductID = Courses.CourseID
    left join course_limits on course_limits.CourseID = Courses.CourseID
```

Dont_make_payment_in_time

Widok ten przedstawia użytkowników, którzy nie zapłacili do wymaganego czasu (nawet jak czas został przedłużony).

```
create view dont_make_payment_in_time as
    select UserID,
           case
            when ExtendedPaymentDeadline is not null then ExtendedPaymentDeadline
            else PaymentDeadline
        end offical_payment_date,
        ProductID
    from Orders_Details inner join Orders on Orders_Details.OrderID = Orders.OrderID
    where
        case
            when ExtendedPaymentDeadline is not null then ExtendedPaymentDeadline
            else PaymentDeadline
        end < getdate() and Payment is null
```

Financial_report

Widok przedstawia raport finansowy z każdego produktu w bazie.

```
create view financial_report as
    with count_products_sales as (
        select Orders_Details.ProductID, sum(Payment) as total_payment
```

```
        from Orders_Details inner join Products on Orders_Details.ProductID = Products.ProductID
        inner join Categories on Products.CategoryID = Categories.CategoryID
        where Name <> 'Reunion'
        group by Orders_Details.ProductID
    )

    select Products.ProductID, Name
    from Products left join Categories on Products.CategoryID = Categories.CategoryID
    left join count_products_sales on Products.ProductID = count_products_sales.ProductID
    where Name != 'Reunion'
```

Future_meetings

Widok przedstawia przyszłe spotkania studyjne.

```
create view future_meetings as
    select *
    from meetings_information
    where getdate() < DateAndBeginningTime
```

Future_meetings_sign

Widok przedstawia ile osób zapisało się na przyszłe spotkania.

```
create view future_meetings_sign as
    with meetings_sign_count as (
        select MeetingID, count(UserID) as sign_count
        from Users_Meetings_Attendance
        group by MeetingID
    )

    select future_meetings.MeetingID, sign_count
    from future_meetings inner join meetings_sign_count on future_meetings.MeetingID = meetings_sign_count.MeetingID;
```

Meeting_sign_limit

Widok przedstawia ilość osób zapisanych na dane spotkanie i jego limit.

```
create view meeting_sign_limit as
    with meeting_limits as (
        select Meetings.MeetingID, min(Limit) as total_limit
        from Meetings inner join In_person_Meetings on Meetings.MeetingID = In_person_Meetings.MeetingID
        group by Meetings.MeetingID
    )

    select Meetings.MeetingID, total_product_orders, total_limit
    from Meetings left join products_orders on Meetings.MeetingID = products_orders.ProductID
    left join meeting_limits on meeting_limits.MeetingID = Meetings.MeetingID
```

Meetings_information

Widok przedstawia ogólne informacje o spotkaniach studyjnych.

```
create view meetings_information as
    select MeetingID,
           FirstName,
           LastName,
           Name,
           ReunionID,
           DateAndBeginningTime,
           Duration,
           Price,
           TypeName
    from Meetings inner join Employees on Meetings.TeacherID = Employees.EmployeeID
    inner join Subjects on Meetings.SubjectID = Subjects.SubjectID
    inner join Types on Meetings.TypeID = Types.TypeID
```

Orders_Payment_informations

Widok przedstawia informacje o szczegółach wszystkich zamówień.

```
create view orders_payment_informations as
    select Orders.OrderID,
           SubOrderID,
           FirstName,
           LastName,
           Orders_Details.ProductID,
           Name,
           case
               when ExtendedPaymentDeadline is null then PaymentDeadline
               else ExtendedPaymentDeadline
           end as offical_payment_deadline,
           FullPrice,
           PaymentDate
    from Orders_Details inner join Orders on Orders_Details.OrderID = Orders.OrderID
    inner join Users on Orders.UserID = Users.UserID
    inner join Products on Orders_Details.ProductID = Products.ProductID
    inner join Categories on Products.CategoryID = Categories.CategoryID
```

Products_orders

Widok przedstawia ile razy każdy produkt został zamówiony.

```
create view products_orders as
    select ProductID, count(SubOrderID) as total_product_orders
    from Orders_Details
    group by ProductID
```

Studie_sign_limit

Widok przedstawia ile osób zapisało się na studia i jaki jest ich limit.

```
create view studie_sign_limit as
    with studie_limits as (
        select Studies.StudiesID, min(Limit) as total_limit
        from Studies inner join Subjects on Studies.StudiesID = Subjects.StudiesID
        inner join Meetings on Subjects.SubjectID = Meetings.SubjectID
        inner join In_person_Meetings on Meetings.MeetingID = In_person_Meetings.MeetingID
        group by Studies.StudiesID
    )

    select Studies.StudiesID, total_product_orders, total_limit
    from Studies left join products_orders on products_orders.ProductID = Studies.StudiesID
    left join studie_limits on studie_limits.StudiesID = Studies.StudiesID
```

Studies_meetings_list

Widok ten przedstawia listę obecności dla spotkań stydujnych wraz z datą odbycia się ich.

```
create view studies_meetings_list as
    select
        Users_Meetings_Attendance.MeetingID,
        format(DateAndBeginningTime, 'yyyy-mm-dd') as date,
        FirstName,
        LastName,
        Present
    from Users_Meetings_Attendance inner join Users on Users_Meetings_Attendance.UserID = Users.UserID
    inner join Meetings on Users_Meetings_Attendance.MeetingID = Meetings.MeetingID
```