

STUDIO TATUAŻU

SEBASTIAN AUGUSTYN

1 Wykaz tabel: 9 tabel

1. Artyści: dane artystów
2. Klienci: dane klientów
3. Rezerwacje: szczegóły rezerwacji
4. Sesje: szczegóły sesji (zrealizowanych rezerwacji)
5. Tatuaze: szczegóły dotyczące usług poszczególnych artystów
6. Urlopy: informacje dotyczące urlopów artystów
7. RabatyOkresowe: informacje o rabatach
8. StatusyRezerwacji: rodzaje statusów rezerwacji
9. StatusyKlienta: rodzaje statusów klienta

2. Wykaz widoków: 2 widoki

1. [v_Przychod]

Zwraca tabelę, w której znajdują się informacje dotyczące tego, ile wynosił przychód w poszczególnych miesiącach.

```
CREATE VIEW [v_Przychod]
AS
SELECT YEAR(DataSesji) AS Rok, MONTH(DataSesji) AS Miesiąc, SUM(KwotaDoZapłaty) AS
Przychód
FROM Sesje
GROUP BY YEAR(DataSesji), MONTH(DataSesji)
```

2. [v_NadchodzaceSesje]

Zwraca tabelę, w której znajdują się wszystkie sesje, które się jeszcze nie odbyły. |

```
CREATE VIEW [v_NadchodzaceSesje]
AS
SELECT Rezerwacje.ID AS RezerwacjaID, Rezerwacje.KlientID, Klienci.Telefon AS
TelefonKlienta, Rezerwacje.DataSesji, StatusyRezerwacji.Nazwa
FROM Rezerwacje
JOIN Klienci ON Rezerwacje.KlientID = Klienci.ID
JOIN StatusyRezerwacji ON Rezerwacje.StatusRezerwacjiID = StatusyRezerwacji.ID
WHERE DataSesji > GETDATE()
```

3 . Wykaz funkcji: 3 funkcje

1. ArtystaZarobek

Po wprowadzeniu zmiennej w postaci ArtystaID zwraca tabelę, w której znajdują się informacje dotyczące łącznej ilości sesji i zarobionych pieniędzy przez artystę, którego dotyczy zmienna.

```
CREATE FUNCTION ArtystaZarobek
    (@ArtystaID INT)
RETURNS TABLE
AS
RETURN

    SELECT
        Artysci.Imie + ' ' + Artysci.Nazwisko AS DaneArtysty,
        COUNT(Sesje.ArtystaID) AS IlośćSesji,
        SUM(Sesje.KwotaDoZapłaty) AS Zarobek
    FROM Artysci
    JOIN Sesje ON Artysci.ID = Sesje.ArtystaID
    WHERE
        Artysci.ID = @ArtystaID
    GROUP BY Artysci.Imie + ' ' + Artysci.Nazwisko
```

2. KwotaNaKlienta

Zwraca średnią kwotę jaką wydała w studiu jedna osoba.

```
CREATE FUNCTION KwotaNaKlienta
    ()
RETURNS INT
AS
BEGIN

    DECLARE @Średnia INT
    SET @Średnia = ( SELECT
        SUM(Sesje.KwotaDoZapłaty)/COUNT(DISTINCT(Sesje.KlientID))
        FROM Sesje)

    RETURN @Średnia
END
```

3. UrlopyPracownikow

Po wprowadzeniu zmiennej w postaci ArtystaID zwraca tabelę, w której znajdują się informacje dotyczące ilości wykorzystanych dni urlopowych i ilości dni urlopowych, które jeszcze może wykorzystać w danym roku artysta, którego dotyczy zmienna.

```
CREATE FUNCTION UrlopyPracownikow
    (@ArtystaID INT)
RETURNS TABLE
AS
RETURN

    SELECT  CAST(YEAR(GETDATE())AS INT) AS Rok,
            Artysci.Imie + ' ' + Artysci.Nazwisko AS DaneArtysty,
            ISNULL(SUM(DATEDIFF(dd, Urlopy.PoczątekUrlopu,
Urlopy.KoniecUrlopu)),0) AS Wykorzystano,
            14 - ISNULL(SUM((DATEDIFF(dd, Urlopy.PoczątekUrlopu,
Urlopy.KoniecUrlopu))),0) AS Zostało
            FROM Artysci
            LEFT JOIN Urlopy ON Artysci.ID = Urlopy.ArtystaID
            GROUP BY
            Artysci.Imie + ' ' + Artysci.Nazwisko, Artysci.ID
            HAVING
            Artysci.ID = @ArtystaID
```

4. Wykaz procedur: 3 procedury

1. DodajRezerwacje

Dodaje rezerwację (nowy rekord) w tabeli Rezerwacje po wprowadzeniu wszystkich potrzebnych informacji. DataRezerwacji jest uzupełniana na podstawie aktualnej daty.

```
CREATE PROCEDURE DodajRezerwacje
(
    @KlientID int,
    @ArtystaID int,
    @DataSesji datetime,
    @CzasSesji time,
    @StatusRezerwacji int
)
AS
BEGIN
    DECLARE @DataRezerwacji datetime
    SET @DataRezerwacji = GETDATE()
    INSERT INTO Rezerwacje VALUES (@KlientID, @ArtystaID, @DataRezerwacji, @DataSesji,
    @CzasSesji, @StatusRezerwacji)
    SELECT * FROM Rezerwacje
END
```

2. DodajNaCzarnaListe

Wszystkim klientom których ID jest powiązane z rezerwacjami, które zostały anulowane (Rezerwacje.StatusRezerwacjiID = 3) zmienia StatusKlienta w tabeli Klienci na 1 (CzarnaLista).

```
CREATE PROCEDURE DodajNaCzarnaListe
```

```
AS
```

```
BEGIN
```

```
    UPDATE Klienci
```

```
    SET Klienci.StatusKlienta = 1
```

```
    FROM Klienci JOIN Rezerwacje ON Klienci.ID= Rezerwacje.KlientID
```

```
    WHERE Rezerwacje.StatusRezerwacjiID = 3
```

```
    SELECT * FROM Klienci
```

```
    WHERE StatusKlienta = 1
```

```
END
```

3. UstalCeny

Aktualizuje kolumnę KwotaDoZapłaty w tabeli Sesje o kwotę, która jest wyliczana na podstawie ceny bazowej tatuażu (Tatuaże.CenaBazowa) pomnożonej przez ilość godzin sesji (Sesje.CzasSesji) oraz rabat (RabatyOkresowe.Procent), jeśli akurat obowiązuje.

```
CREATE PROCEDURE UstalCeny
```

```
AS
```

```
BEGIN
```

```
    UPDATE Sesje
```

```
    SET KwotaDoZapłaty= DATEPART(hh, Sesje.CzasSesji) * Tatuaże.CenaBazowa *  
(CAST(100 - ISNULL(RabatyOkresowe.Procent, 0)AS FLOAT)/100)
```

```
    FROM Sesje
```

```
    JOIN Tatuaże ON Sesje.ArtystaID = Tatuaże.ArtystaID
```

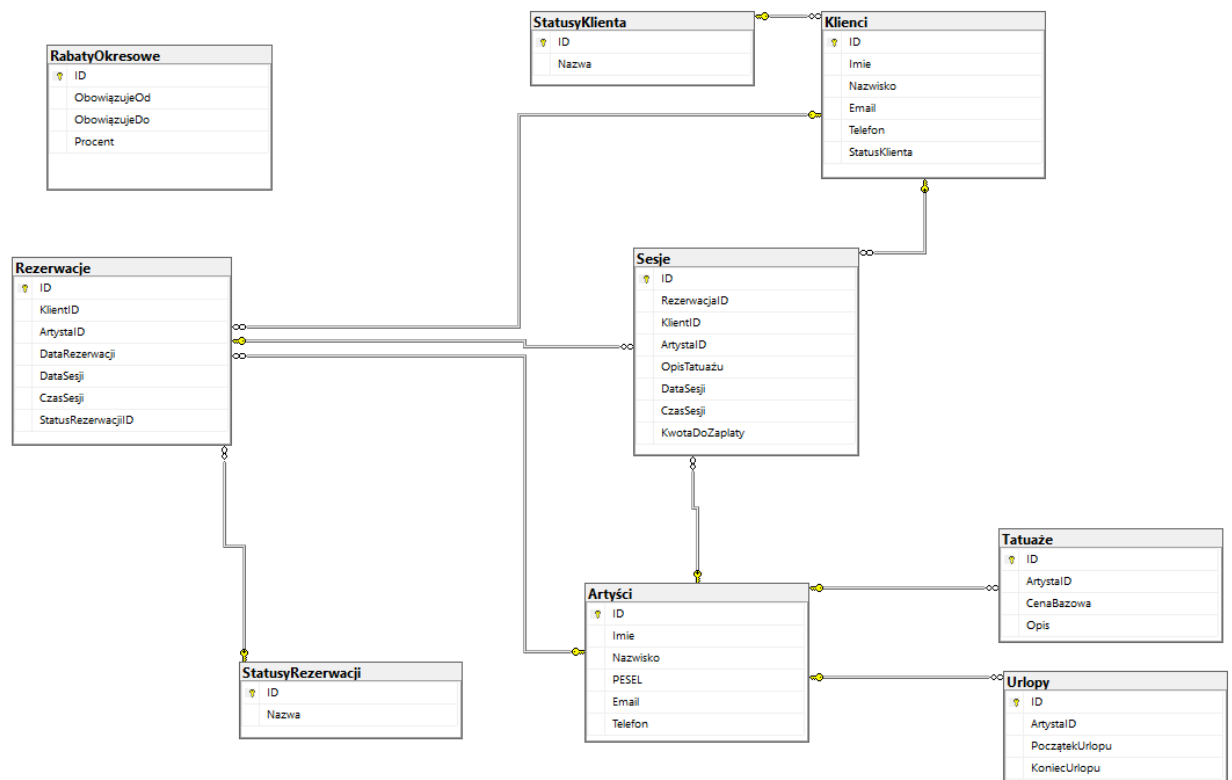
```
    LEFT JOIN RabatyOkresowe ON Sesje.DataSesji BETWEEN RabatyOkresowe.ObowiązujeOd
```

```
AND RabatyOkresowe.ObowiązujeDo
```

```
    SELECT * FROM Sesje
```

```
END
```

5.Diagram ERD:



6.Informacje dodatkowe:

Przed sprawdzaniem przykładowych instrukcji proszę wywołać procedurę UstalCeny.

EXEC UstalCeny

Pozwoli to na pełne zaprezentowanie instrukcji, które korzystają z `Sesje.KwotaDoZaplaty`.

7.Przykładowe instrukcje:

1.W studiu znaleziono czapkę.

Sprawdzenie do którego klienta może należeć, razem z jego danymi kontaktowymi.

```
SELECT Klienci.Imie + ' ' + Klienci.Nazwisko AS DaneKlienta, Klienci.Telefon
FROM Klienci
JOIN Sesje ON Klienci.ID = Sesje.KlientID
WHERE Sesje.DataSesji BETWEEN '2021-10-17' AND '2021-10-20'
```

2.Wyświetlenie najczęściej wykonywanego wzoru/wzorów.

```
SELECT TOP 1 WITH TIES OpisTatuazu, COUNT(OpisTatuazu) AS IleRazyWykonany
FROM Sesje
GROUP BY OpisTatuazu
ORDER BY COUNT(OpisTatuazu) DESC
```

3. Wyświetlenie ile w danym miesiącu zarobiło studio, wraz z informacją, czy odnotowało przy tym zysk czy stratę.

```
SELECT YEAR(DataSesji) AS Rok, MONTH(DataSesji) AS Miesiąc, SUM(KwotaDoZapłaty) AS  
Przychód,  
CASE  
WHEN SUM(KwotaDoZapłaty) > '3000' THEN 'Zysk'  
ELSE 'Strata'  
END AS Wynik  
FROM Sesje  
GROUP BY YEAR(DataSesji), MONTH(DataSesji)  
ORDER BY YEAR(DataSesji) DESC, MONTH(DataSesji) ASC
```

4. Wyświetlenie klienta, który wydał najwięcej oraz kwoty, którą wydał.

```
SELECT  
TOP 1 WITH TIES Klienci.Imie + ' ' + Klienci.Nazwisko AS DaneKlienta,  
SUM(Sesje.KwotaDoZapłaty) AS Zapłacono  
FROM Klienci  
JOIN Sesje ON Klienci.ID = Sesje.KlientID  
GROUP BY Klienci.Imie + ' ' + Klienci.Nazwisko  
ORDER BY SUM(Sesje.KwotaDoZapłaty) DESC
```

5. Wyświetlenie ile godzin w 2021 roku przepracowali poszczególni artyści, wraz z ich danymi.

```
SELECT  
Artyści.Imie + ' ' + Artyści.Nazwisko AS DaneArtysty,  
SUM(DATEPART(HOUR, (Sesje.CzasSesji))) AS CzasPracy  
FROM Artyści  
JOIN Sesje ON Artyści.ID = Sesje.ArtystaID  
WHERE YEAR(DataSesji) = 2021  
GROUP BY Artyści.Imie + ' ' + Artyści.Nazwisko
```

6. Wyświetlenie przyszłych sesji, danych kontaktowych klientów powiązanych z danymi sesjami oraz instrukcji związanych ze statusem rezerwacji.

```
SELECT Rezerwacje.ID, Rezerwacje.KlientID, Klienci.Email, Klienci.Telefon,  
CASE  
WHEN StatusRezerwacjiID = 2 THEN 'WyślijPrzypomnienie'  
WHEN StatusRezerwacjiID = 3 THEN 'DodajDoCzarnejListy'  
ELSE 'WyślijZaproszenie'  
END AS StatusRezerwacji  
FROM Rezerwacje  
JOIN Klienci ON Rezerwacje.KlientID = Klienci.ID  
WHERE Rezerwacje.DataSesji > GETDATE()
```

7. Wyświetlenie ilości tatuaży wykonanych w konkretnych stylach, pogrupowanych ze względu na rok wykonania tatuażu. Pozwala prognozować czy dany styl nabiera na popularności, czy wręcz odwrotnie.

```
SELECT  
YEAR(Rezerwacje.DataSesji) AS Rok,  
Tatuaże.Opis AS Styl,  
COUNT(Rezerwacje.ArtystaID) AS IlośćRezerwacji  
FROM Rezerwacje  
JOIN Tatuaże ON Rezerwacje.ArtystaID = Tatuaże.ArtystaID  
GROUP BY YEAR(DataSesji), Tatuaże.Opis  
ORDER BY Tatuaże.Opis DESC, YEAR(DataSesji) DESC
```

8. Wyświetlenie rezerwacji, które nie doszły do skutku.

```
SELECT Rezerwacje.ID, Rezerwacje.KlientID, Rezerwacje.ArtystaID,  
Rezerwacje.DataRezerwacji, Rezerwacje.DataSesji  
FROM Rezerwacje  
LEFT JOIN Sesje ON Rezerwacje.ID=Sesje.RezerwacjaID  
WHERE Sesje.RezerwacjaID IS NULL AND Rezerwacje.DataSesji < GETDATE()
```