

Opis procedur i funkcji dostępnych dla poszczególnych typów użytkowników :

Zwykły pracownik :

Podczas wywoływania dostępnych procedur korzysta z parametru @prac , który jest typu Integer oraz oznacza indeks pracownika. **ZWYKŁY PRACOWNIK NIE MOŻE GO ZMIENIAĆ !**

- Liczba_godzin_pracownika(@prac) : Funkcja zwraca łączną sumę godzin pracownika oraz korzystając z relacji wyświetla ile godzin tygodniowo pracownik pracuje dla poszczególnego zespołu.
- Wynagrodzenie_pracownika(@prac) : Funkcja zwraca łączną sumę wynagrodzenia pracownika oraz korzystając z relacji wyświetla ile pracownik zarabia podczas pracy w poszczególnym zespole.
- Zespoły_pracownika(@prac) : Procedura wyświetla wszystkie zespoły do, których należy oraz należał pracownik wraz ze swoim obecnym statusem w nich.
- Projekty_pracownika(@prac) : Procedura wyświetla wszystkie projekty , które realizuje oraz realizował pracownik wraz z obecnym statusem tych projektów.
- Dane_pracownika(@prac) : Procedura wyświetla dane pracownika , w przypadku błędu w danych należy skontaktować się z odpowiednią instancją.
- Historia_pracownika(@prac) : Procedura wyświetla wszystkie zespoły oraz realizowane przez nich projekty wraz ze statusem projektów. (Jest to przydatne w momencie , kiedy pracownik np. : ubiega się o dołączenie do nowego zespołu w firmie i wraz z podaniem wysła wygenerowaną przez aplikację historie).

Zarządca zespołu :

Należy pamiętać o tym , że zarządca zespołu także jest pracownikiem => może korzystać z funkcjonalności aplikacji jak zwykły pracownik. Natomiast z powodu posiadanych obowiązków dostaje także nowe możliwości. Zadaniem zarządcy zespołu jest nadzorowanie pracy określonego zespołu. Podczas wywoływania swoich procedur korzysta z parametrów : @prac - do funkcjonalności dla zwykłego pracownika , @zesp - jest to parametr typu Integer oraz oznacza indeks zespołu , @cz - jest to parametr typu Integer oraz oznacza indeks członka zespołu, @wyn - Nie ujemny integer określający wysokość wynagrodzenia dla poszczególnego członka , @czas - Nie ujemny integer określający czas pracy. Zarządca może ustalić wartość parametru @cz , @wyn oraz @czas jednak **NIE MOŻE ZMIENIAĆ PARAMETRÓW @prac i @zesp!**

- Budżet_zespołu(@zesp) : Funkcja zwraca łączny budżet zespołu , który jest sumą przydzielonych budżetów do wszystkich projektów realizowanych przez zespół.

- Członkowe_zespołu(@zesp) : Procedura korzystając z tabeli Grupy_zespołów wyświetla wszystkich pracowników , którzy znajdują się w danym zespole.
- Ustal_wynagrodzenie(@zesp,@cz,@wyn) : Procedura modyfikuje tabele Grupy_zespołów, do atrybutu "wynagrodzenie" wpisuje wartość @wyn w krotce gdzie atrybut "zespół" = @zesp oraz atrybut "pracownik" = @prac
- Ustal_czas_pracy_członka(@zesp,@czas,@wyn) : Procedura modyfikuje tabele Grupy_zespołów, do atrybutu "godziny" wpisuje wartość @czas w krotce gdzie atrybut "zespół" = @zesp oraz atrybut "pracownik" = @prac.
- Ustal_czas_pracy(@zesp,@czas) : Procedura modyfikuje tabele Zespoły , do atrybutu "godziny_tygodniowo" wpisuje wartość @czas w krotce gdzie atrybut "zespół" = @zesp.
- Zatwierdź_wynagrodzenie(@zesp) : Funkcja działa jako transakcja do atrybutu "wypłata" w tabeli Pracownik dodaje wartość atrybutu "wynagrodzenie" z tabeli Grupy_zespołów , wszędzie tam gdzie Pracownik.id_pracownika = Grupy_zespołów.pracownik i Grupy_zespołów.zespol = @zesp. Jeżeli jednak suma wynagrodzeń przekracza budżet zespołu transakcja kończy się niepowodzeniem.
- Zatwierdź_czas_pracy(@zesp) : Funkcja działa jako transakcja do atrybutu "godziny_tygodniowo" w tabeli Pracownik dodaje wartość atrybutu "godziny" z tabeli Grupy_zespołów , wszędzie tam gdzie Pracownik.id_pracownika = Grupy_zespołów.pracownik i Grupy_zespołów.zespol = @zesp. Jeżeli jednak suma godzin przekracza budżet zaplanowana sumęgodzin pracy dla całego zespołu transakcja kończy się niepowodzeniem.
- Dodaj_do_zespołu(@zesp,@cz) : Procedura dodaje do zespołu pracownika o indeksie @cz. Atrybut "status" w tabeli Grupy_zespołów w krotce gdzie "pracownik" = @cz oraz "zespol" = @zesp zmienia się na "Obecny". (W przypadku dodania nie odpowiedniej osoby należy skontaktować się z odpowiednią instancją aby poprawić zmiany w historii).
- Usuń_z_zespołu(@zesp,@cz) : Procedura usuwa z zespołu pracownika o indeksie @cz. Atrybut "status" w tabeli Grupy_zespołów w krotce gdzie "pracownik" = @cz oraz "zespol" = @zesp zmienia się na "Usunięty". (W przypadku usunięcia nie odpowiedniej osoby należy skontaktować się z odpowiednią instancją aby poprawić zmiany w historii).
- Przenieś_z_zespołu(@zesp,@cz) : Procedura usuwa z zespołu pracownika o indeksie @cz. Atrybut "status" w tabeli Grupy_zespołów w krotce gdzie "pracownik" = @cz oraz "zespol" = @zesp zmienia się na "Przeniesiony". (W przypadku przeniesienia nie odpowiedniej osoby należy skontaktować się z odpowiednią instancją aby poprawić zmiany w historii).

- Projekty_zespołu(@zesp) : procedura wyświetla wszystkie projekty w , których bierze bądź brał udział zespół @zesp.

Zarządca działu :

Należy pamiętać o tym , że zarządca działu jest także pracownikiem => może korzystać z funkcjonalności aplikacji jak zwykły pracownik. Natomiast z powodu posiadanych obowiązków dostaje także nowe możliwości. Zadaniem zarządcy działu jest nadzorowanie pracy określonego działu poprzez zarządzanie zleceniami i zespołami. Ponadto zarządca działu może sprawdzać oraz modyfikować wewnętrzne ustawienia zespołu. Podczas wywoływania swoich procedur korzysta z parametrów : @prac - do funkcjonalności dla zwykłego pracownika , @zesp - jest to parametr typu Integer oraz oznacza indeks zespołu , @cz - jest to parametr typu Integer oraz oznacza indeks członka zespołu, @wyn - Nie ujemny integer określający wysokość wynagrodzenia dla poszczególnego członka , @czas - Nie ujemny integer określający czas pracy , @kl - Nie ujemny integer określający id klienta, @dost - varchar określający nazwę dostawcy, @nazwa - varchar(90) określający nazwę , @wart - Integer określający wartość , @zl - nie ujemny integer określający id zlecenia , @pr - nie ujemny integer określający id projektu , @dz - nie ujemny integer określający id działu, @st - varchar(90) określający status projektu. Zarządca może ustalić wartość parametru @zesp, @cz , @wyn , @czas , @kl, @dost ,@nazwa, @wart, @zl ,@pr , @st jednak **NIE MOŻE ZMIENIAĆ PARAMETRÓW @prac i @dz!**

- Dodaj_klienta(@nazwa) : Procedura dodaje do tabeli Klienci nowy wiersz gdzie Klienci.nazwa_klienta = @kl_nazwa.
- Dodaj_zlecenie(@kl,@wart,@dz) : Procedura dodaje do tabeli Zlecenia nowy wiersz gdzie Zlecenia.klient = @kl , Zlecenia.dzial = @dz oraz Zlecenia.wartosc_zlecenia = @wart_zl.
- Zakup_produkt(@nazwa, @dost) : Procedura dodaje produkt o nazwie @nazwa, i łączy go z dostawcą o nazwie @dost (dodając takiego jeśli nie istnieje w tabeli)
- Utworz_projekt(@nazwa,@zl,@zesp) : Procedura dodaje do tabeli Projekty nowy wiersz gdzie Projekty.zlecenie = @zl oraz Projekty.nazwa_projektu = @nazwa oraz Projekty.zespol = @zesp. Procedura nie wykonuje się jeżeli zespol o indeksie @zesp nie należy do działu.
- Ustal_budzet_projektu(@pr , @wart,@dz) : Procedura modyfikuje tabele Projekty , w miejscu gdzie "id_projektu" = @pr w atrybucie "przydzielony_budzet" wpisuje wartosc @wart.Procedura nie wykonuje się jeżeli zlecenie przydzielone do projektu (Projekt.zlecenie nie należy do działu => porównuje odpowiadający atrybut Zlecenia.dzial z @dz w miejscu gdzie

Zlecenia.id_zlecenia = Projekty.zlecenie i Projekty.id_projektu = @pr)

- Sprawdz_bilans(@dz) : Funkcja wyświetla wszystkie zlecenia podjęte przez dział oraz ich wartość, wszystkie projekty realizowane przez dział oraz ich budżety wraz z zespołami które je realizują oraz wszystkie zespoły znajdujące się w dziale oraz ich budżety. Następnie od sumy wszystkich zleceń odejmuje sumę budżetów wszystkich zespołów i pokazuje wynik.
- Zleczenie_info(@zl ,@dz) : Procedura jeżeli Zlecenia.dział = @dz wyświetla z tabeli Zlecenia w miejscu gdzie "id_zlecenia" = @zl atrybut "wartosc_zlecenia" oraz "klient" oraz z tabeli Projekty w miejscu gdzie Projekty.zlecenie = @zl wszystkie informacje z tabeli.
- Zmien_status(@pr , @st, @dz) : Procedura modyfikuje tabelę Projekty w miejscu gdzie atrybut "id_projektu" = @pr ustawia atrybut "status" na wartość @st. Jeżeli projekt jest wykonywany przez zespół , którego dział = @dz.
- Zatwierdz_budzet(@dz) : Funkcja działa jako transakcja , modyfikuje wartości tabeli Zespoly.budzet ustawiając sumę z wartości z Projekty.przydzielony_budzet wszędzie tam gdzie Zespoly.id_zespolu = Projekty.zespol.
- ZM_zarzadcy_zespolu(@zesp,@cz,@dz) : Procedura zmienia status obecnego zarządcy zespołu o indeksie @zesp na "Były zarządca" oraz ustanawia członka zespołu o indeksie @cz nowym zarządcą zmieniając mu status na "Zarządca". Procedura nie wykonuje się jeżeli dział zespołu != @dz.
- Utworz_zespol(@cz,@nazwa,@dz) : Procedura dodaje nową wiersz w tabeli Zespoły , do atrybutu "zarządca_zespołu" przypisuje wartość @cz, do atrybutu "nazwa_zespołu" przypisuje wartość @nazwa, do atrybutu "dział" przypisuje wartość @dz.
- Rozwiaz_zespol(@zesp,@dz) : Procedura jeżeli zespół o indeksie @zesp należy do działu @dz zmienia status wszystkich projektów zespołu o indeksie @zesp na "Porzucony" oraz przenosi wszystkich członków zespołu @zesp.

Szef :

Jest to osoba z najwyższym poziomem dostępu równym administratorowi bazy danych. Zatrudnia i zwalnia pracowników oraz ma możliwość korygowania błędów.

Podczas wywoływania swoich procedur korzysta z parametrów :

@prac - do funkcjonalności dla zwykłego pracownika , @zesp - jest to parametr typu Integer oraz oznacza indeks zespołu , @cz - jest to parametr typu Integer oraz oznacza indeks członka zespołu, @wyn - Nie ujemny integer określający wysokość wynagrodzenia dla poszczególnego członka , @czas - Nie ujemny integer określający czas pracy , @kl - Nie ujemny integer określający id klienta, @nazwa

- varchar(90) określający nazwę , @wart - Integer określający wartość , @zl - nie ujemny integer określający id zlecenia , @pr - nie ujemny integer określający id projektu , @dz - nie ujemny integer określający id działu , @st - varchar(90) określający status projektu , @atr - varchar(90) określający atrybut w tabeli , @nowy - varchar(90) określający nową wartość atrybutu. **MOŻE USTALIĆ WARTOŚĆ KAŻDEGO PARAMETRU !**

- Korekta_danych(@prac,@atr, @nowy) : Procedura modyfikuje wartości w tabeli Pracownik w miejscu gdzie "id_pracownika" = @prac . Procedura modyfikuje atrybut @atr.
- Korekta_historii(@prac,@zesp,@atr, @nowy) : Procedura modyfikuje wartości w tabeli Grupy_zespołów w miejscu gdzie "pracownik" = @prac oraz "zesp" = @zesp, modyfikuje atrybut @atr (zazwyczaj powinien być to status) istnieje też możliwość wpisania w miejsce @atr "Delete" wtedy procedura usuwa wiersz a tabeli.
- Dodaj_dział(@prac, @nazwa) : Procedura dodaje do tabeli Działy nowy dział , gdzie wartość atrybutu zarządca_działu = @prac oraz nazwa_działu = @nazwa.
- Zwolnij_pracownika(@prac) : Procedura usuwa z tabel Loginy i Hasła krotki gdzie id_pracownika = @prac . Ponadto zmienia status pracownika o indeksie @prac we wszystkich projektach na "Były pracownik".
- ZM_zarządcy_działu(@dz , @prac) : Procedura modyfikuje tabele Działy zmieniając atrybut zarządca_działu na @prac w miejscu gdzie id_działu = @dz.
- Przeniesienie_zlecenia(@zl, @dz) : Procedura działa transakcyjnie, zmienia status wszystkich projektów związanych z zleceniem @zl na "Zamknięty" oraz redukuje ich budżety do zera. Następnie modyfikuje tabele Zlecenia gdzie id_zlecenia = @zl zmieniając wartość dział na @dz.
- Zamknij_dział(@dz) : Procedura działa transakcyjnie , wywołuje procedure Przeniesienie_zlecenia dla każdego zlecenia w dziale @dz (podczas uruchomienia wyświetla się okno pytające do , którego działu przenieść zlecenie). Następnie usuwa wszystkich pracowników z zespołów które należą do @dz (Szef może wywoływać procedury zarządcy zespołu). Następnie zmienia zarządcę działu na 0.