PROJEKT ZALICZENIOWY

# Forma:

1. Projekt Visual Studio SSDT (baza danych)
2. Projekt wgrany do repozytorium GitHub wraz z historią Commit’ów.
3. Skrypty ładujące dane do utworzonej bazy jako:
   1. Procedury
   2. Pre-Execute Script w projekcie SSDT (dodatkowo punktowane)

# Uwagi ogólne:

1. Dobierz **odpowiednie typy danych**
2. W całej bazie stosuj **spójne typy tekstowe** (Regular lub Unicode)
3. W nazwach Tabel i Kolumn **nie używaj** spacji, polskich znaków i znaków specjalnych {@,#,$,%,^,&,\*,.,...}, nie zaczynaj nazw kolumn od liczb
4. Stosuj **spójną konwencję nazewnictwa** (Pascal/Camel Case; język PL/ENG)
5. W skryptach ładujących dane do tabel **NIE UŻYWAĆ** kluczy technicznych – doczytać na temat INSERT/UPDATE/DELETE+JOIN

# CZĘŚĆ (A)

Zaprojektuj bazę danych dla salonu samochodowego.

Baza powinna zawierać następujące tabele:

1. Marki – marki samochodów sprzedawanych w salonie
2. Modele – modele samochodów sprzedawanych w salonie
3. Cennik – cennik modeli samochodów. cennik powinien zawierać dane historyczne – każdy wiersz powinien zawierać model samochodu, cenę, oraz daty obowiązywania (od, do).
4. Pracownik – pracownicy zatrudnieni w salonie
5. Klient – tabela klientów salonu samochodowego
6. Zamówienia – tabela zawierająca wszystkie zamówienia złożone w salonie samochodowym, powinna zawierać następujące informacje:
   1. Klient, który złożył zamówienie
   2. Pracownik, u którego złożono zamówienie
   3. Model zamówionego samochodu
   4. Cena Sprzedaży
   5. Data zamówienia, Data planowanego odbioru
   6. Informacja o tym, czy zamówienie zostało już zrealizowane (typ BIT 0/1)
   7. Informacja o tym, czy zamówienie zostało już opłacone (typ BIT 0/1)

Wymagania:

1. Samodzielnie zaproponuj informacje, które powinny być przechowywane w tabelach (tzn. dobierz kolumny).
2. Każda tabela powinna posiadać klucz główny (PK). Klucz techniczny (surrogate key) utworzony za pomocą IDENTITY.
3. Utwórz odpowiednie klucze obce (FK). Klucze obce powinny mieć włączoną opcję ON DELETE/UPDATE w trybie NULL lub DEFAULT. W trybie DEFAULT zaproponuj znacznik wartości nie znanej dla kolumn wchodzącyuch w skład klucza – np. -1 dla wartości liczbowych.
4. W tabeli Pracownicy dodaj kolumnę PESEL, upewnij się, że w tabeli nie ma duplikatów (UNIQUE)
5. W tabeli Klienci dodaj kolumnę Nr. Dowodu Osobistego, upewnij się, że w tabeli nie ma duplikatów (UNIQUE)
6. W tabeli Zamówienia upewnij się, że cena sprzedaży jest większa od zera (CHECK)

# CZĘŚĆ (B)

Rozbuduj bazę danych o następujące obszary:

1. Obszar HR
   1. Słownik stanowisk pracy
   2. Słownik zespołów
   3. Przypisanie pracownika do zespołu (M2M)
   4. Przypisanie pracownika do stanowiska w danym okresie (M2M – pracownik, stanowisko, okres od/do)
   5. Tabela z wynagordzeniem pracownika w danym okresie (pracownik, kwota, okres od/do)
   6. Tabela z wypłatami (pracownik, kwota, typ -> pensja/premia, okres za który wypłacono)
2. Obszar Serwis
   1. Tabela usług oferowanych przez serwis
   2. Zamówienia
   3. Status zamówienia (słownik) – wymyślić listę statusów w jakich może być zamówienie wraz z kolejnością np. złożone, w realizacji, czeka na odbiór, zamknięte…
   4. Historia zamówień – po jednym wpisie dla każdego statusu, data ustawienia, pracownik, który zmienił status

# CZĘŚĆ (B)

Rozbuduj bazę danych o następujące obszary:

1. Obszar HR
   1. Słownik stanowisk pracy
   2. Słownik zespołów
   3. Przypisanie pracownika do zespołu (M2M)
   4. Przypisanie pracownika do stanowiska w danym okresie (M2M – pracownik, stanowisko, okres od/do)
   5. Tabela z wynagordzeniem pracownika w danym okresie (pracownik, kwota, okres od/do)
   6. Tabela z wypłatami (pracownik, kwota, typ -> pensja/premia, okres za który wypłacono)
2. Obszar Serwis
   1. Tabela usług oferowanych przez serwis
   2. Zamówienia
   3. Status zamówienia (słownik) – wymyślić listę statusów w jakich może być zamówienie wraz z kolejnością np. złożone, w realizacji, czeka na odbiór, zamknięte…
   4. Historia zamówień – po jednym wpisie dla każdego statusu, data ustawienia, pracownik, który zmienił status

# CZĘŚĆ (C)

W bazie danych utwórz schemat o nazwie [rapOp]. W schemacie [rapOp] utwórz widoki zwracające następujące wyniki:

**Widoki:**

1. Lista wszystkich marek wraz z informacją o ilości zamówień samochodów danej marki oraz sumaryczną kwotą sprzedaży
2. Lista wszystkich zamówień wraz z informacją o udzielonym rabacie (cena katalogowa obowiązująca w danym okresie minus cena sprzedaży dzielone przez cenę katalogową)
3. Lista modeli samochodów, które nie pojawiają się na żadnym z zamówień (nie zostały ani razu zamówione)
4. Lista pracowników, wraz z informacją o tym ile zamówień zrealizowali oraz łączną kwotą tych zamówień

# CZĘŚĆ (D)

W schemacie [rapOp] utworzyć po 3 widoki raportowe dla każdego z obszarów HR/Serwis. Zaproponować samodzielnie, opisać (np. w komentarzu w widoku) co prezentują raporty. We wszystkich widokach wymusić poziom izolacji READ UNCOMMITED.

# CZĘŚĆ (E)

1. Wstawić dane do wszystkich tabel oprócz zamówień (Salon/Serwis).
2. Przygotować prodedury (USP), które wygenerują dowolną ilość zamówień. Procedury mają wstawiać losowe ale sensowne dane. (przyda się: WHILE, RAND)

# CZĘŚĆ (F)

1. Zdenormalizować model:
   1. Przygotować drugą bazę danych o zdenormalizowanych strukturach. Spróbować zbliżyć się do modelu gwiazdy wokół tabel Zamówienie.
   2. Przygotować procedury, które pobierają dane z pierwszej (OLTP) i ładują je do bazy zdenormalizowanej (OLAP).
   3. Założyć odpowiednie klucze główne i obce.
   4. Na tabelach zamówień założyć indeksy COLUMNSTORE.

# CZĘŚĆ (G)

W zdenormalizowanej bazie utworzyć schemat [rapAn]. W schemacie [rapAn] utworzyć raporty zbiorcze. We wszystkich widokach wymusić poziom izolacji READ UNCOMMITED. Samodzielnie zaproponować treść prezentowaną przez raporty.