

Wymagania projektowe:

Projekt będzie oceniany w czterech kategoriach.

Prezentacja (5 pkt.)

W niniejszej kategorii ocenianie będzie to jak Państwo przedstawia swój pomysł. W szczególności, ocenie podlega:

- Wybór odpowiedniego problemu do rozważenia w projekcie. Oceniane będzie to czy pomysł pozwala na wypełnienie wymagań dotyczących jakości technicznej bazy (np. odpowiednia ilość tabel). Czasem już na poziomie planowania widać, że pomysł nie jest odpowiedni do stworzenia projektu.
- Wypełnianie terminowo dodatkowych wymagań, o których wspomniane będzie na końcu dokumentu (terminarz)
- Zrozumienie tematu. Wybierając jakiś pomysł trzeba zastanowić się nad różnymi edge case'ami. Przykład: Jeżeli planujecie bazę PKP to trzeba sobie odpowiedzieć np. na pytanie: "Czy pracownik x może być jednocześnie w pociągu y i z?" lub "Czy klient może kupić bilet na dwa pociągi jednocześnie, na tę samą porę?". Nie jest to problemem, że dokonacie pewnych uproszczeń, problemem będzie to, jeżeli nie jesteście ich świadomi.

Projekt bazy (12 pkt.)

W tej kategorii oceniony zostanie techniczny aspekt bazy danych. Można powiedzieć, że traktuje ona o "szkieletcie projektu". W szczególności ocenie podlega:

- Odpowiednia ilość tabel. Powinno ich być 8-12, z tym, że zależnie od problemu może ich być mniej (to bardzo mało prawdopodobne) albo więcej. W szczególności, ilość tabel **musi** być skonsultowana ze mną (zostanie to dokonane poprzez regularne wysyłanie materiałów, o które będę prosił)
- Poprawne użycie kluczy głównych, obcych czy wszelakich więzów integralności. Dla przykładu: imię i nazwisko nie są dobrymi kandydatami na klucz główny. Jeżeli kolumna przechowuje ceny to potrzebuje założenia odpowiedniego ograniczenia dziedziny itp.
- Baza powinna zawierając logicznie zaplanowane tabele, uwzględniające takie rzeczy jak relacje wiele-do-wielu czy dynamiczną enumerację. Czy zawsze trzeba używać dynamiczną enumerację zamiast statyczną (np. poprzez check)? Nie, chociaż najczęściej opłaca się użyć właśnie dynamicznej enumeracji.
- Oprócz tabel powinny znaleźć się odpowiednie widoki. Najczęściej widoki reprezentują "raportowanie" danych z bazy. Przykład: "Lista x najlepszych sprzedawców w miesiącu y".
- (* - dla ciekawskich) Profesjonalne podejście do projektu wymaga czasem pewnych mniej standardowych rozwiązań. Klasycznym przykładem jest archiwizacja danych finansowych. Innym przykładem (na pozór podobnym do archiwizacji) jest bi-temporalizacja danych (dość zaawansowany

temat), gdzie informacje o okresie "aktualności" danych jest podawana dwuwymiarowo. Pierwszy wymiar mówi o tym, kiedy były zmiany na bazie, a drugi mówi o tym w jakim okresie dana miała jaką wartość. Przykład: dzisiaj wiemy o tym, że informacja posiada cechę "X" o wartości "A" i pod tym względem dokonujemy pewnej decyzji. Tak naprawdę tego dnia informacja zmieniła swoją cechę "X" na wartość "B", ale tego nie wiedzieliśmy, dowiedzieliśmy się o tym następnego dnia. Nasza decyzja była oparta jedynie na wiedzy, którą posiadaliśmy. Dzięki bi-temporalizacji możemy "obronić" naszą decyzję, zaznaczając to kiedy mieliśmy jaką wiedzę o obiekcie. Dla chętnych mogę opowiedzieć o tym dokładniej. Jeszcze innym aspektem jest przemyślenie tematu zakładania indeksów na bazie. Przede wszystkim wtedy, gdy na bazie istnieje archiwizacja danych, która odbywa się poprzez trzymanie informacji w jednej tabeli. Przy tym pomyśle mamy problem: po jakimś czasie w tabeli znajduje się >90% obserwacji nieaktualnych. Stąd wszelkie zapytania muszą przejść przez bardzo dużą (procentowo) ilość danych, które wiemy, że nas nie interesują. Rozwiązaniem tego problemu są warunkowe indeksy, które indeksują podzbiór danych w tabeli. Dla przykładu, niech w tabeli `tbl` jest kolumna `isActive` trzymająca informację o tym czy dana krotka jest aktywna (możliwe wartości to `TRUE/FALSE`). Stąd przy zakładaniu indeksów można zrobić `WHERE isActive = TRUE` i wtedy indeksowane są JEDYNNIE dane, które mają `TRUE` na `isActive`. Innymi słowy, przy wykonywaniu `SELECT`, jeżeli uwzględnimy w `WHERE isActive = TRUE`, to wtedy przy przeglądaniu danych z tabeli, olewane są te miejsca w pamięci, które zawierają nieaktualne dane. No dobra, Ci którzy zrozumieli to co napisałem to szacun. Krótkie wyjaśnienie. Przeszukiwanie danych w bazie (które są gdzieś w pamięci) to coś podobnego jak szukanie informacji w jakiejś książce. Oczywiście możemy przeglądać strona po stronie i sprawdzać czy znajdziemy to czego potrzebujemy, jednakże jest to dość średni pomysł. Lepszym pomysłem jest wybranie się na koniec książki i zagłębienie do indeksu (zbieżność nazw nie jest przypadkowa). Tak samo jest z wykonywaniem operacji na bazie - zakłada się indeksy na tabeli aby przeszukiwanie było szybsze. Taki indeks informuje gdzie krotka zawierająca daną informację jest w pamięci. W przypadku indeksów warunkowych stworzony jest indeks jedynie na pewnym podzbiórze danych. Dzięki takiej technice nie tracimy na czasie wykonywania operacji.

Interfejs graficzny (8 pkt.)

Uwaga! Zanim opowiem co będzie oceniane w tej kategorii chciałbym zaznaczyć jedną rzecz. Interfejs graficzny jest **warunkiem koniecznym** zaliczenia projektu. Proszę nie traktować tego jako: "luz, możemy mieć z tego 0 punktów". W tej kategorii będę oceniał warstwę graficzną projektu. Interfejs może być napisany w dowolnym języku programowania. Ja będę pokazywał jak napisać przykładową aplikację w R za pomocą pakietu `shiny`. Dodatkowe zajęcia odbędą się na początku stycznia. W szczególności oceniane będzie:

- Idea aplikacji jest następująca: ktoś kto się **nie zna na SQLu** powinien umieć obsługiwać interfejs. Innymi słowy "ukrywamy", że po wciśnięciu przycisku "wyślij" jest wykonywane zapytanie po stronie bazy danych.
- Realizacja prostych zapytań typu `select/update/delete`. W szczegól-

ności powinna być możliwość dodania/aktualizacji informacji poprzez uzupełnienie odpowiednich formularzy. Zapytania typu `select` powinny być realizowane, przede wszystkim, poprzez widoki.

- Estetyka kodu będzie oceniana tylko "pozytywnie". To jest, jeżeli kod będzie brzydki jak Manchester United to trudno, natomiast jeżeli będzie piękny jak Liverpool FC no to będzie można zarobić dodatkowe punkty. Podobnie jeżeli chodzi o aspekty graficzne interfejsu.
- Nawiązując do poprzedniego podpunktu. Ważne jest aby interfejs działał, także aplikacja może być brzydka ale niech nie ma błędów.

Dodatki (5 pkt.)

W ostatniej kategorii oceniane są wszelkie dodatkowe rzeczy dołożone do projektu. W szczególności funkcje, triggery, ciekawe rozwiązania, rozbudowane widoki. Dodatkowo, można wykorzystać ciekawsze typy danych jak np. JSON. Jest to dość indywidualna sprawa, także można przegadać tę kategorię poprzez maile/konsultacje. Jest to też kategoria, która niejako miesza się z poprzednimi, ale w niej będzie oceniane wyjście poza standardowe rozwiązania.

Kalendarz wysłania dodatkowych materiałów:

Do 20.12.2020: Proszę wysłać informację zawierającą spis tabel wraz z kolumnami oraz typami tych kolumn. Może to być w wykonanie w dowolnej formie, w szczególności możecie wpisać to do treści maila.

Do 03.01.2021: Proszę o wysłanie diagramu zawierającego informację o bazie wraz z relacjami pomiędzy tabelami. Przykładowy taki diagram znajduje się na Moodlu w pliku "schemat finalnej bazy ze skryptu listing_w05.sqlPlik" (19 listopada). W najbliższym czasie (a więc przez 20 grudnia) prześlę informację jak to zrobić najłatwiej.

Do 10.01.2021: Proszę o informację (szczegółową) jak Państwo będą generowali dane do bazy. W szczególności, jeżeli przewidujecie losowanie danych za pomocą jakiegoś języka programowania to chciałbym zobaczyć skrypty wykonujące losowanie.

Do 17.01.2021: Proszę o dostarczenie skryptów zakładających szkielet bazy danych. W osobnym pliku powinny znaleźć się pliki zakładające tabele. W następnym więzy integralności (można też dla wygody rozdzielić zakładanie np. kluczy obcych oraz innych więzów). W kolejnych plikach powinny znaleźć się definicje widoków, w następnym definicje funkcji/triggerów itp. Idea jest taka, że wszystkie składowe bazy danych mają znajdować się w osobnych plikach z rozszerzeniem .sql. W zależności od sposobu generowania danych zostaniecie poproszeni o dosłanie danych w jakiejś formie.

Do 24.01.2021: Proszę o informację o interfejsie graficznym. W szczególności jaki jest stan prac, jak interfejs wygląda (mogą być screeny), jaki język jest użyty itp. Będzie to ponad tydzień przed prezentacjami, stąd interfejs może być jeszcze w budowie (choć nie polecam).

Do 30.01.2021: Niezależnie od terminu prezentacji jest to **ostateczny** termin oddania wszelkich materiałów do bazy danych. Po tym terminie nie będę niczego już przyjmował. To co doślecie do tego dnia + prezentacja będzie oceniane jako

Wasz projekt bazy.

Uwaga nr 1! Może się zdarzyć, że wyślecie coś terminowo ale poproszę Was o dosłanie czegoś jeszcze. Nie będzie to negatywnie oceniane, macie prawo się pomylić. Jednakże **30.01.2021** jest ostatecznym terminem na dosyłanie czegokolwiek.

Uwaga nr 2! Jest czym totalnie normalnym, że możecie nie mieć na coś pomysłu. Tym bardziej, jeżeli chodzi o kwestie czysto programistyczne (np. generowanie danych). Ważne jest to abyście odpowiednio wcześniej odezwali się do mnie. Jakoś to razem rozwiążemy. Jednakże proszę uważać. Jeżeli weźmiecie się za wszystko pod koniec stycznia olewając powyższe terminy, to moja ochota do pomocy gwałtownie spadnie ;)