

# Instrukcja do projektu BD2

## (wersja C, semestr letni)

Tomasz Traczyk

Semestr 22L

W czasie trwania semestru instrukcja może być sukcesywnie uzupełniana i poprawiana (zmiany będą oznaczone **kolorem**). Należy zawsze pracować z aktualną wersją instrukcji. O zauważonych błędach czy brakach w instrukcji proszę informować prowadzącego przedmiot Tomasza Traczyka.

## 1 Wprowadzenie

### 1.1 Cel projektu

Celem projektu BD2 jest przećwiczenie projektowania struktury relacyjnej bazy danych dla prostego problemu OLTP, z zużyciem modelowania ER (*Entity-Relationship*).

### 1.2 Etapy projektu

Projekt jest podzielony na etapy:

1. Koncepcja rozwiązania: na podstawie treści zadania tworzony jest szkic modelu ER struktury danych; koncepcja ta jest uzgadniana z prowadzącym.
2. Model ER: na podstawie uzgodnionej koncepcji i treści zadania tworzony jest diagram związków encji (ERD).
3. Projekt i generowanie struktur danych: wyniki modelowania ER są przekształcane w model danych relacyjnych, który następnie jest udoskonalany. Na podstawie projektu generowane są pliki DDL, służące do stworzenia struktur w bazie danych. Struktury danych są wypełniane danymi testowymi. Opcjonalnie dodawane są uzupełniające elementy projektu (wyzwalacze, dodatkowe ograniczenia itp.).

### 1.3 Odbiory etapów projektu i terminarz

Odbiory poszczególnych etapów projektu odbywają się „wirtualnie”, przy użyciu systemu Studia i poczty elektronicznej. Wymagane pliki przekazuje się za pomocą modułu „Sprawozdania” systemu Studia. Wszelkie pytania i wątpliwości należy zgłaszać za pomocą poczty elektronicznej do osób prowadzących odpowiednie części projektu.

**Uwaga:** do nadawania e-maili proszę korzystać wyłącznie z serwerów pocztowych na PW. W miarę możliwości proszę używać adresów e-mail zawierających nazwisko nadawcy. Listy wysyłane z innych serwerów oraz podpisane różnymi „ksywkami” mogą być ignorowane! We wszelkiej korespondencji dotyczącej projektu proszę **w tytule maila podawać numer zadania**, a tam gdzie to adekwatne także nazwę konta Oracle.

### 1.3.1 Terminy oddawania etapów projektu

Etap	Temat	Termin
1.	Koncepcja rozwiązania	03-28 godz. 10:00
	Koncepcja rozwiązania – poprawki	04-11 godz. 10:00
2.	Model ERD	04-25 godz. 10:00
	Model ERD – poprawki	05-09 godz. 10:00
3.	Projekt struktur	05-30 godz. 10:00
	Projekt struktur – poprawki	06-15 godz. 24:00

## 1.4 Samodzielność, jakość i ocena prac

Prace projektowe muszą być wykonane całkowicie samodzielnie. Niedozwolone jest korzystanie z „osiągnięć” poprzedników, źródeł internetowych, odbywanie konsultacji z kolegami itp. Stwierdzenie niesamodzielności pracy skutkuje niezaliczeniem projektu, bez możliwości poprawki.

Podawaną w tabelach w dalszej części instrukcji ważność poszczególnych pozycji należy rozumieć następująco: wymagania opisane jako „Wymagane” muszą być spełnione pod rygorem niezaliczenia! Niespełnienie wymagań opisanych jako „Zalecane” powoduje obniżenie oceny, zaś spełnienie wymagań opisanych jako „Opcjonalne” może powodować podwyższenie oceny.

Prace świadczące o ignorancji autora, tj. zawierające błędy szczególnie rażące lub w bardzo dużej liczbie, będą oceniane na 0. Niedopuszczalne są usiłowania zaliczenia projektu „metodą prób i błędów”.

Sprawdzenie i pozytywne ocenienie przez prowadzącego wyników danego etapu projektu jest niezbędnym warunkiem przystąpienia do etapu kolejnego. W kolejnym etapie obowiązuje poprawienie wszystkich błędów wskazanych przez prowadzącego; brak poprawy błędów skutkuje oceną 0.

Na sprawdzonych pracach umieszczana jest ogólna konkluzja.

**Zaliczone** jeśli praca nie wymaga wykonania poprawki (wskazane usterki należy poprawić w kolejnym etapie).

**Do poprawki** jeśli praca wymaga wykonania poprawki.

**Projekt niezaliczony** jeśli z wyżej wymienionych przyczyn praca została oceniona negatywnie bez możliwości poprawki.

Od etapu 2. na pracach umieszczana jest także ocena punktowa. Znaczenie poszczególnych ocen częściowych podane jest w opisie etapów.

## 2 Uwagi techniczne

### 2.1 Środowisko wykonywania projektu

Projekt jest wykonywany na własnym sprzęcie uczestników, ale konieczne za pomocą oprogramowania Oracle SQL Developer Data Modeler 20.4 lub nowszego.

**Baza danych** Struktury danych przedstawione do sprawdzenia muszą być (wraz z danymi przykładowymi) umieszczone w udostępnionej przez Wydział dydaktycznej bazie danych. Dostęp do tej bazy jest możliwy także spoza Uczelni, dane połączenia są następujące:

- Host: ora2.ia.pw.edu.pl
- Port: 1521
- SID/Service name: iaais

**Konta uczestników zajęć** Każdemu uczestnikowi „z urzędu” zakładane jest konto na ww. serwerze Oracle iais; konto to ma nazwę BD2C###, gdzie ### jest przydzielonym uczestnikowi numerem (lista z numerami będzie udostępniona na serwerze Studia).

**Uwaga:** w dalszej części instrukcji notacja BD2C### będzie oznaczała tę właśnie nazwę z przydzielonym numer uczestnika.

## 2.2 Problemy techniczne

Wszelkie problemy techniczne należy niezwłocznie zgłaszać e-mailem do Tomasza Traczyka.

Zgłoszenie musi zawierać szczegółowy opis problemu i okoliczności jego powstania, w tym:

- nazwę i hasło użytkownika Oracle,
- kody (literowo-cyfrowe) błędów,
- komunikaty o błędach (w całości).

Zgłoszenia nie zawierające dostatecznie szczegółowej informacji o zaistniałym problemie nie będą rozpatrywane.

**Uwaga:** problemy, które nie zostały zgłoszone, nie będą mogły stanowić usprawiedliwienia ewentualnego opóźnienia wykonania części projektu.

## 2.3 Kwestie techniczne

**Kopie rezerwowe plików projektowych** Dane z bazy danych nie są backupowane! Dlatego w swym dobrze pojętym interesie każdy uczestnik projektu powinien we własnym zakresie wykonywać kopie rezerwowe informacji wytworzonych w czasie projektu.

**Sesje użytkownika** Ze względu na ograniczone zasoby serwera uprasza się nie utrzymywać dużej liczby jednocześnie otwartych sesji połączenia z bazą Oracle. Należy unikać uruchamiania dużej liczby programów równocześnie i zamykać narzędzia w danej chwili niepotrzebne (np. zbędne sesje SQL Developera).

## 2.4 Konfiguracja środowiska

Oprogramowanie niezbędne do realizacji projektu, tj. programy Oracle SQL Developer Data Modeler (wersja 20.4 lub nowsza) oraz Oracle SQL Developer, może być legalnie ściągnięte z sieci, ze strony Oracle<sup>1</sup>.

Prowadzący projekt mają bardzo ograniczone możliwości udzielania konsultacji dotyczących instalacji i uruchamiania oprogramowania na komputerach studentów, do czynności tych nie należy zatem zabierać się w ostatniej chwili.

**Sprawdzenie dostępu do bazy Oracle** Przed przystąpieniem do etapu 3 wszyscy powinni zawnazasu sprawdzić działanie konta na serwerze Oracle.

1. Sprawdzić dostęp do serwera Oracle: wywołać program Oracle SQL Developer. Stworzyć połączenie do bazy, używając przydzielonej nazwy użytkownika Oracle (BD2C###) i takiego samego hasła. Uwaga: hasła są wrażliwe na wielkość liter! W razie problemów niezwłocznie powiadomić pocztą elektroniczną Tomasza Traczyka.
2. Zmienić hasło na serwerze Oracle (zalecane): używając polecenia **password** lub zdania SQL **alter user ... identified by ...**; zmienić hasło na serwerze Oracle.

## 3 Etap 1: Koncepcja rozwiązania

Ten etap ma umożliwić uzgodnienie rozumienia zadania z prowadzącym.

<sup>1</sup>Wymagana jest rejestracja, ale nie niesie ona żadnych złych konsekwencji ani nie powoduje spamu.

**Rozdanie zadań** Zadania będą rozesłane za pomocą mechanizmu Sprawozdań w systemie Studia, znajdą się w sprawozdaniu pt. *Etap 1: ERD*.

**Konsultacje zrozumienia treści zadania** Zalecane jest zweryfikowanie rozumienia treści zadania przez skonsultowanie się z prowadzącym. Pytania dotyczące treści zadania należy wysłać e-mailem do Tomasza Traczyka. **Proszę w tytule maila podawać numer zadania.**

Uwaga: w ramach konsultacji prowadzący nie będzie dokonywał wstępnej oceny całych diagramów, zatem oczekiwane są pytania bardziej szczegółowe niż *Czy to jest dobrze?*

### 3.1 Szkic koncepcji rozwiązania

Na tym etapie wykonać należy pokazujący rozumienie zadania szkic modelu koncepcyjnego w postaci diagramu ER w notacji Barkera.

#### 3.1.1 Wymagania ogólne

Nazwa	Opis	Ważność
Zgodność z zadaniem	Model danych powinien odpowiadać treści zadania i pokrywać cały zakres zadania. Nie jest wskazane rozszerzające interpretowanie treści zadania.	Wymagane
Czytelność	Model musi być czytelny i łatwy do zrozumienia, co osiągnąć można m.in. przez stosowanie czytelnych nazw, zgodnych z terminologią użytą w zadaniu.	Wymagane
Stosowanie zaleceń projektowych	Tworząc model należy stosować się do zaleceń projektowych i wskazówek podanych na wykładzie.	Wymagane
Poprawna notacja	Model musi poprawnie używać notacji Barkera.	Wymagane

Szkic modelu – jak sama nazwa wskazuje – może być niekompletny, ale nie może być niepoprawny, np. rażąco niezgodny z zaleceniami projektowymi i wskazówkami podanymi na wykładzie, narysowany w niewłaściwej notacji itp.

#### 3.1.2 Artefakty

Szkic modelu danych powinien zawierać następujące artefakty.

Nazwa	Opis	Ważność
Encje	Encje odzwierciedlające całą strukturę danych, jaką autor ma zamiar zaprojektować.	Wymagane
Atrybuty	Wymagane są jedynie najważniejsze atrybuty, niezbędne dla zrozumienia koncepcji. Nie trzeba jeszcze określać typów danych.	Wymagane
Związki	Związki powinny dobrze odzwierciedlać istotę rozumienia zadania; w przypadkach wątpliwych warto związki nazwać.	Wymagane
Identyfikatory	Wskazanie identyfikatorów (PUID) jest zalecane <b>jeśli są one potrzebne do zrozumienia koncepcji</b> .	Zalecane
Komentarze	Komentarze (notatki) umieszczone na diagramie mogą być użyte do wyjaśnienia szczególnych zawiłości (ale proszę nie nadużywać!).	Opcjonalne

Instrukcja do tworzenia diagramów ERD znajduje się w części 6.1.

### 3.2 Odbiór etapu 1

**Prowadzący odbiór** Ten etap sprawdza Tomasz Traczyk.

**Przesłanie diagramu** Diagram ERD należy przekazać w postaci pliku PDF.

1. Sformatować diagram ERD na 1 stronę wydruku: encje na diagramie ustawić tak, by cały diagram mieścił się na jednej stronie A4 w ustawieniu pionowym (*portrait*), zajmując na tyle dużą część strony, by wszystkie napisy dało się wygodnie przeczytać bez użycia lupy! Wygenerowany plik PDF ma odpowiadać takiej jednej stronie. Diagramy nie dające się wydrukować na jednej stronie A4 oraz nieczytelne będą odrzucane.
2. Diagram zaopatrzyć w dane twórcy i zadania: na diagramie, w jednym z górnych rogów, umieścić jako notatkę dane twórcy i zadania: numer konta Oracle w formacie **BD2C###**, gdzie **###** jest numerem konta użytkownika w bazie Oracle, numer zadania, nazwisko i imię oraz adres e-mail autora (w miarę możliwości proszę używać adresów zawierających nazwisko nadawcy). Diagramy bez takiego opisu będą odrzucane!
3. Uporządkować diagram: sprawdzić czy wszystkie istotne napisy są widoczne, np. nie zostały obcięte przez zbyt małe rozmiary symboli encji. Diagramy powinny dać się wydrukować na drukarce monochromatycznej, nie należy więc nadużywać kolorów, w szczególności ciemnych kolorów wypełnienia; najlepiej wypełnienie ustawić na przezroczyste lub białe.
4. Wygenerować plik PDF z diagramem, zgodnie z instrukcją z części 6.1.
5. Wygenerowany plik obejrzeć przed wysłaniem!
6. Plik z diagramem przekazać za pomocą systemu Studia jako sprawozdanie **Etap 1: ERD**.

#### Uwagi

- Przesyłki nie zawierające wymaganych informacji będą odrzucane.
- Wszelkie pytania proszę przysyłać e-mailem (nie przez Teams!). **Proszę w tytule maila podawać numer zadania.**

### 3.3 Ocena i poprawianie etapu 1

Etap 1 nie jest oceniany „na punkty”, ale musi się koniecznie skończyć akceptacją koncepcji przez prowadzącego. Bez takiej akceptacji nie można przystąpić do etapu 2!

Sprawdzone prace będą **umieszczone w systemie Studia**. Jeśli pracę opatrzone adnotacją „Do poprawki”, obowiązuje poprawienie koncepcji i terminowe przesłanie do ponownego sprawdzenia. Przysługuje jedna poprawka (w szczególnych przypadkach dwie, o ile starczy czasu), przy czym prawo do poprawki jest uzależnione od terminowości przesłania wyniku danego „podejścia”: prace spóźnione mogą nie być dopuszczone do poprawiania ze względu na brak czasu.

**Forma odbioru poprawek** Diagram ERD przygotować jak poprzednio. Plik z diagramem przekazać za pomocą systemu Studia jako sprawozdanie **Etap 1: ERD-poprawka**.

Kolejne poprawki – jeśli ich zażądano – należy przysyłać niezwłocznie, w tej samej formie co poprzednio, e-mailem bezpośrednio do osoby która dokonywała oceny pracy. **Proszę w tytule maila podawać numer zadania.**

**Uwzględnienie uwag prowadzącego** W każdym przypadku uwagi prowadzącego muszą być uwzględnione w czasie realizacji kolejnego podejścia lub etapu. W tym celu niezbędne jest zapoznanie się ze sprawdzoną pracą – każda sprawdzona praca jest odsyłana! Jeśli jednak pracy nie opatrzone adnotacją „Do poprawki”, poprawek nie należy przysyłać, lecz uwagi uwzględnić w kolejnym etapie.

## 4 Etap 2: Model ERD

Ten etap ma określać wymagania co do budowanego systemu danych w sposób umożliwiający zaprojektowanie systemu. Etap kolejny – projekt – powinien dać się wykonać na podstawie wiedzy zgromadzonej w czasie tego etapu i wykonanego teraz modelu, bez konieczności uzyskiwania dodatkowych informacji źródłowych.

### 4.1 Kompletny model ERD

Na tym etapie wykonać należy kompletny model logiczny w postaci diagramu związków encji dla problemu opisanego w zadaniu.

#### 4.1.1 Szczegółowy model związków encji – wymagania

Model związków encji powinien mieć stopień szczegółowości umożliwiający przekształcenie do wstępnego projektu tabel za pomocą narzędzia. Wyniki tego etapu powinny zatem spełniać m.in. następujące wymagania:

Nazwa	Opis	Ważność
Zgodność z zadaniem	Model ERD powinien być poprawionym i uzupełnionym modelem uzgodnionym w etapie 1. Nie należy tworzyć modelu znacząco różniącego się od uzgodnionego.	Wymagane
Czytelność	Model musi być czytelny i łatwy do zrozumienia, co osiągnąć można m.in. przez stosowanie czytelnych nazw, zgodnych z terminologią użytą w zadaniu.	Wymagane
Stosowanie zaleceń projektowych	Tworząc model należy stosować się do zaleceń projektowych i wskazówek podanych na wykładzie.	Wymagane
Model <b>logiczny</b>	Model ma być w formie gotowej do przekształcenia na projekt struktur relacyjnych (identyfikatory zdatne do przekształcenia na klucze obce, związki $n - m$ rozbite na związki $1 - n$ itd.).	Wymagane
Poprawna notacja	Model musi poprawnie używać notacji Barkera.	Wymagane
Prawidłowe użycie narzędzia	Model należy stworzyć za pomocą narzędzia Oracle SQL Developer Data Modeler, właściwie wykorzystując możliwości tego narzędzia.	Wymagane

#### 4.1.2 Artefakty

W wyniku analizy powinny zostać wytworzone następujące artefakty:

Nazwa	Opis	Ważność
Encje i atrybuty	Definicje wszystkich encji przewidzianych w systemie.	Wymagane
Właściwości encji	Określenie właściwości encji, w tym <b>Name</b> , <b>Short Name</b> , <b>Preferred Abbreviation</b> .	Wymagane
Atrybuty	Definicje wszystkich atrybutów wszystkich encji.	Wymagane
Podstawowe właściwości atrybutów	Pełne określenie podstawowych właściwości atrybutów, w tym: nazwa, typ i długości, opcjonalność.	Wymagane
Związki	Definicje wszystkich związków.	Wymagane

...

Nazwa	Opis	Ważność
Właściwości związków	Określenie właściwości związków, w tym stopnia i opcjonalności po obu stronach oraz – jeśli trzeba – nieprzenośności (nietransferowalności) po stronie „wiele”.	Wymagane
Właściwe przygotowanie związków	Związki $n - m$ rozbite na związki $1 - n$ ; związki $1 - 1$ nie mają prawa wystąpić.	Wymagane
Nazwy związków	Związki nazwane (najlepiej obustronnie) w sposób umożliwiający zrozumienie ich znaczenia.	Zalecane
Identyfikatory	Definicje identyfikatorów, umożliwiające wytworzenie – w wyniku automatycznej transformacji – definicji odpowiednich kluczy.	Wymagane
Pierwotne unikalne identyfikatory	Pierwotny unikalny identyfikator (PUID) zdefiniowany dla każdej encji i spełniający wymagania.	Wymagane
Wtórne unikalne identyfikatory	Wtórne unikalne identyfikatory zdefiniowane jeśli dodatkowa kontrola niepowtarzalności jest potrzebna.	Zalecane
Dokumentacja encji i atrybutów	Opisy encji i atrybutów (we właściwościach <b>Comment in RDBMS</b> ), stanowiące dokumentację zaprojektowanej struktury.	Zalecane

Uwaga: Nie używać konwencji nazw **CośTam**, lecz **Coś tam** (ze spacjami).

## 4.2 Odbiór etapu 2

**Prowadzący odbiór** Ten etap sprawdza Tomasz Traczyk.

### Przesłanie diagramu

1. Zweryfikować diagram ERD (zalecane): sprawdzić diagram ERD pod względem kompletności, czytelności, poprawności nazw itp.
2. Przygotować diagram do przesłania, sformatować i przekształcić na plik PDF jak w etapie 1 (patrz 3.2)
3. Plik z diagramem przekazać za pomocą systemu Studia jako sprawozdanie Etap 2: ERD.

### Przesłanie raportu

1. Wygenerować plik z pomocniczym raportem: wybrać pozycję menu File / Reports. W polu Available Reports wybrać Entities, wyłączyć obie opcje Embed... Włączyć opcję Generate HTML TOC in separate file. Skasować wszystkie wygenerowane pliki oprócz AllEntitiesDetails\_\*.html.
2. Ten plik przed wysłaniem przeczytać w celu weryfikacji poprawności i kompletności (typy danych, opisy dla tabel i kolumn).
3. Plik przekazać za pomocą systemu Studia jako sprawozdanie Etap 2: raport.

Uwagi:

- Przesyłki nie zawierające wymaganych informacji będą odrzucane.
- Ewentualne pytania proszę wysłać e-mailem na adres prowadzącego (nie przez Teams).

### 4.3 Ocena i poprawianie etapu 2

Z etapu 2 można otrzymać do 20 punktów (na 40 punktów z całego projektu), po 10 dla kategorii: Z – zgodność z zadaniem, P – poprawność modelu ER.

Sprawdzone prace będą **umieszczone w systemie Studia**. Jeśli pracę opatrzone adnotacją „Do poprawki”, obowiązuje jej poprawienie. Przysługuje jedna poprawka (w wyjątkowych przypadkach dwie), przy czym prawo do poprawek może być uzależnione od terminowego przekazania pierwszego „podejścia” do etapu. Poprawkę mogą także przesłać inne osoby „na ochotnika”, ale tylko jeśli liczba uzyskanych punktów nie przekroczyła limitu podanego przez prowadzącego. Poprawkę należy przesłać w podanym terminie.

**Forma odbioru poprawek** Pliki przygotować jak poprzednio i przekazać za pomocą systemu Studia jako sprawozdania Etap 2: ERD-poprawka i Etap 2: raport-poprawka.

Kolejne poprawki – jeśli ich zażądano – należy przysyłać niezwłocznie, w tej samej formie co poprzednio, e-mailem bezpośrednio do osoby która dokonywała oceny pracy.

**Uwzględnienie uwag prowadzącego** W każdym przypadku uwagi prowadzącego muszą być uwzględnione w czasie realizacji kolejnego podejścia lub etapu. W tym celu niezbędne jest zapoznanie się ze sprawdzoną pracą. Każda sprawdzona praca jest odsyłana. Jeśli jednak pracy nie opatrzone adnotacją „Do poprawki”, poprawek nie należy przysyłać, lecz uwagi uwzględnić w kolejnym etapie.

### 4.4 Najczęstsze błędy w etapie 2

Poniższa tabela zestawia najczęstsze błędy popełniane w modelu ERD. Podane tu kody błędów będą używane podczas sprawdzania prac.

Kod	Opis
U-NSPEC	Zbyt mało specyficzna nazwa atrybutu identyfikującego (np. tylko ID).
U-NAZ	Nazwy nie nadają się na pierwotny UID, bo są za długie i zmienne.
U-BRAK	Brak UID.
U-NADM	PUID błędny formalnie lub nadmiarowy (np. związek identyfikujący + sztuczny UID).
U-NM	Błędny PUID w encji realizującej zw. $n - m$ .
N-CS	Nie należy stosować nazw typu 'jakieśCoś', lecz 'Jakiś coś', gdyż nazwy w bazach danych zwykle nie są <i>case-sensitive</i> .
N-TERM	Użycie błędnej terminologii (tabele, kolumny, klucze, relacja zamiast związek itp.).
N-POWT	W nazwach atrybutów nie należy powtarzać nazwy encji (wyjątkiem są atrybuty identyfikujące).
N-IDENC	W nazwach atrybutów identyfikujących należy użyć nazwy encji (np. dla encji 'Coś' użyć nazwy 'Id czegoś').
N-NM	Zła nazwa encji realizującej związek wiele do wielu (np. mechaniczne sklejenie nazw encji).
N-LANG	Nazwy muszą być gramatycznie dopasowane do języka, np. <i>Customer Id</i> , ale <i>Id klienta</i> .
O-ENC	Brak opisu (komentarza) encji lub zbyt skąpy opis.
O-ATR	Brak opisów (komentarzy) atrybutów lub zbyt skąpe opisy.
T-INT	Ostrzeżenie: w Oracle nie ma typu Integer, a domyślne odwzorowanie na typ natywny jest błędne; nie ma też typów String i Boolean.

...



Kod	Opis
T-PREC	Identyfikatory liczbowe nie powinny być o więcej niż 1..2 rzędy większe od spodziewanego zakresu liczb.
T-NAPIS	Numery domów, telefonów, NIP, PESEL itp. to nie są liczby, lecz napisy złożone (na ogół) z cyfr, należy więc stosować typy znakowe.
T-CHAR	Do napisów o stałej długości (np. NIP, PESEL) należy stosować typy o stałej długości, np. <b>CHAR</b> .
T-KWOT	Kwoty pieniężne, kursy walut, ceny itp. muszą być wyrażane stałoprzecinkowo.
T-MAX	Brak maks. długości napisu lub precyzji liczby.
T-CHAR1	Klasyfikację sztywną należy zwykle realizować jako <b>CHAR(1)</b> .
E-OB1	Związek błędnie lub niepotrzebnie obowiązkowy po stronie 1.
E-ZWOPC	Odwroćenie opcjonalności związku.
E-NTRANS	Nietransferowalność oznaczona po stronie 1 zamiast po stronie „wiele”.
E-NMZLE	Zła realizacja rozwinięcia związku wiele do wielu.
E-NMBRAK	Brak rozwinięcia związku wiele do wielu.
E-DESKR	Brak przynajmniej jednego obowiązkowego deskryptora.
E-REKUR	Związek rekurencyjny obowiązkowy po którejś stronie.
E-ATROPC	Atrybuty niosące rzeczywistą informację nie mogą wszystkie być opcjonalne.
E-ZLOZ	Nie wolno stosować atrybutów złożonych, np. Adres.
E-ODWR	Związek narysowany odwrotnie.

## 5 Etap 3: Projekt i implementacja struktury relacyjnej

W tym etapie ma być stworzony kompletny projekt struktur relacyjnych wraz z elementami uzupełniającymi, np. perspektywami lub wyzwalaczami. Struktury te mają znaleźć się we wskazanej bazie danych. Następnie mają być utworzone dane testowe, które zostaną użyte do wykonania zestawu zdań SQL testujących stworzone struktury.

### 5.1 Projekt struktur danych

Projekt struktur danych relacyjnych ma być utworzony na podstawie modelu ERD, z uwzględnieniem uwag sprawdzającego etap 2. Projekt ten należy zweryfikować oraz dokonać niezbędnych poprawek i uzupełnień.

Model związków encji uzyskany na końcu poprzedniego etapu powinien mieć stopień szczegółowości umożliwiający przekształcenie do wstępnego projektu tabel za pomocą narzędzia. Projektowanie polegać zatem będzie na użyciu tego narzędzia, a następnie na weryfikacji i udoskonaleniu wyników jego działania.

Model związków encji należy zatem przekształcić na projekt struktury relacyjnej zgodnie z instrukcją podaną w części 6.2. Następnie za pomocą *Data Modelera* należy przejrzeć, zweryfikować i poprawić projekt, tak by spełnił on podane niżej wymagania oraz zalecenia projektowe nauczane na wykładzie. W szczególności należy zwrócić uwagę na poprawność nazw obiektów i typów danych.

Do poprawionego projektu relacyjnego należy utworzyć „model fizyczny” i w nim zapisać pozostałe niezbędne decyzje projektowe, zgodnie z instrukcją z części 6.2.

#### 5.1.1 Wymagania ogólne

Nazwa	Opis	Ważność
Uwzględnienie zaleceń oceniającego etap 2	Przed przystąpieniem do tego etapu należy model związków encji zmodyfikować tak, by uwzględnił on zalecenia sprawdzającego. Skuteczność tej poprawy stanowi część oceny etapu!	Wymagane
Projekt dostosowany do wybranego DBMS	Szczegóły projektu mają być dostosowane do użytego DBMS Oracle.	Wymagane
Czytelność	Model musi być czytelny i łatwy do zrozumienia, co osiągnąć można m.in. przez stosowanie czytelnych nazw i zwięzłych opisów.	Wymagane
Stosowanie zaleceń projektowych	Tworząc projekt należy stosować się do zaleceń projektowych podanych na wykładzie.	Wymagane
Prawidłowe użycie narzędzia	Model należy stworzyć za pomocą narzędzia Oracle SQL Developer Data Modeler, właściwie wykorzystując wybrane możliwości tego narzędzia.	Wymagane

Uwaga: modelu z podtypami nie wolno przekształcać w strukturę z łukami i związkami 1 – 1.

### 5.1.2 Artefakty

W wyniku projektowania powinny zostać wytworzone następujące artefakty.

Nazwa	Opis	Ważność
Tabele	Definicje wszystkich potrzebnych tabel i określenie ich właściwości: nazwa, skrót nazwy.	Wymagane
Kolumny	Definicje wszystkich kolumn i określenie ich właściwości: nazwa, typ danych i długość (ew. precyzja i skala liczby), opcjonalność/obowiązkowość, ew. wartość domyślna, ew. ograniczenie <b>check</b> dla kolumny i/lub wartości dopuszczalne. Należy zastosować odpowiednio wyłącznie własne typy Oracle, a nie typy ze standardu SQL. W szczególności w Oracle nie ma typu <b>INTEGER</b> , a typ <b>NUMBER</b> bez podanej precyzji jest zmiennoprzecinkowy!	Wymagane
Więzy <i>primary key</i>	Definicje PK dla każdej z tabel: nazwa, kolumny składowe (w odpowiedniej kolejności!).	Wymagane
Więzy <i>unique key</i>	Definicje dodatkowych UK: nazwa, kolumny składowe (w odpowiedniej kolejności!).	Zalecane
Więzy <i>foreign key</i>	Definicje potrzebnych FK: nazwa, kolumny składowe (w kolejności zgodnej z odpowiednim PK), modyfikowalność, przenaszalność (transferowalność), ew. przynależność do łuku.	Wymagane
Więzy <i>check</i>	Definicje więzów <i>check</i> – jeśli potrzebne (np. do walidacji łuków, wymuszenia wielkich liter itp.): nazwa, warunek.	Zalecane
Indeksy	Definicje niezbędnych indeksów, np. na klucze obce: nazwa, unikalność, lista kolumn (w odpowiedniej kolejności). Indeksy nie powinny się dublować!	Wymagane
Użytkownik pomocniczy	Definicja użytkownika <b>BD2C###_APP</b> – użytkownika „końcowego” korzystającego z danych.	Wymagane
Prawa dostępu	Określenie praw dostępu do tabel dla użytkownika „końcowego”.	Wymagane

...

Nazwa	Opis	Ważność
Komentarze do tabel i kolumn	Krótkie opisy tabel i kolumn we właściwości <i>Comment in RDBMS</i> . Komentarze te powinny być pisane tak, jak gdyby były przeznaczone dla użytkownika b.d., a nie dla programisty/dewelopera.	Zalecane

Uwaga: ponieważ nazwy obiektów (tabel, kolumn, indeksów, więzów itd.) w bazie danych nie są wrażliwe na wielkość liter, nie używać konwencji nazw CoTo, lecz Co\_to (z podkreśleniami). Nazwy muszą spełniać wymagania dla nazw w bazie Oracle, tj. m.in. nie zawierać spacji i znaków specjalnych oraz zaczynać się od litery. Nazwy tabel, kolumn, więzów itp. nie powinny też zawierać znaków narodowych. Nazwy kolumn kluczy obcych nie powinny być prefiksowane nazwą tabeli wskazywanej.

## 5.2 Generowanie struktur danych

### 5.2.1 Struktury danych – artefakty

W tej fazie projektu powinny zostać wytworzone następujące artefakty.

Nazwa	Opis	Ważność
Skrypt tworzący struktury danych	Skrypt tworzący tabele wraz z ograniczeniami i indeksy oraz nadający uprawnienia do obiektów.	Wymagane
Struktury utworzone w bazie danych	Struktury utworzone w bazie danych przez wykonanie wygenerowanych skryptów.	Wymagane

### 5.2.2 Elementy uzupełniające strukturę danych

Do zaprojektowanej struktury danych można stworzyć dodatkowe elementy uzupełniające, takie jak sekwencje, wyzwalacze i perspektywy. Celem jest uzupełnienie struktury danych o mechanizmy automatycznego wypełniania kluczy sztucznych, ograniczenia proceduralne, zabezpieczenia dostępu itp.

Nazwa	Opis	Ważność
Synonimy	Definicje synonimów w narzędziu <i>Data Modeler</i> i synonimy utworzone w bazie danych.	Wymagane
Sekwencje	Definicje sekwencji w narzędziu <i>Data Modeler</i> i sekwencje utworzone w bazie danych.	Zalecane
Mechanizmy wypełniające sztuczne klucze główne	Wyzwalacze lub inne mechanizmy podstawiające wartości z odpowiednich sekwencji do kolumn sztucznych kluczy głównych.	Zalecane
Wyzwalacze wygenerowane przez narzędzie	Wyzwalacze wygenerowane przez narzędzie <i>Data Modeler</i> , np. wstawiające wartości z sekwencji lub nadzorujące nietransferowalność kluczy obcych, i utworzone w bazie danych. Uwaga: wyzwalacze wygenerowane za pomocą narzędzia <i>Data Modeler</i> , należy przed zainstalowaniem w bazie danych ręcznie poprawić zgodnie z zaleceniami podanymi na wykładzie.	Opcjonalne

...

Nazwa	Opis	Ważność
Inne wyzwalacze	Wyzwalacze realizujące inne funkcje, np. ograniczenia proceduralne. Uwaga: proszę nie przesadzać, a w szczególności proszę zwrócić uwagę na to, by za pomocą wyzwalaczy nie realizować funkcjonalności, które daje się osiągnąć za pomocą ograniczeń deklaratywnych i/lub odpowiednio zaprojektowanej struktury danych.	Opcjonalne
Komentarze do programów	Kod elementów proceduralnych (np. wyzwalaczy) wyczerpująco skomentowany, w tym podany cel tworzenia danego elementu.	Zalecane
Perspektywy	Definicje perspektyw, np. ograniczających dostęp do danych użytkownikowi BD2C###_APP, w narzędziu <i>Data Modeler</i> i utworzone w bazie danych.	Opcjonalne

### 5.3 Testowanie struktur danych

Zaprojektowaną i wypełnioną danymi strukturę danych należy przetestować za pomocą odpowiednio skonstruowanych zdań SQL, realizujących zadawanie zapytań i próby naruszania więzów.

#### 5.3.1 Dane testowe – artefakty

W tej fazie projektu powinny zostać wytworzone następujące artefakty.

Nazwa	Opis	Ważność
Dane testowe	Dane testowe wpisane w utworzone struktury w bazie danych.	Wymagane
Skrypt tworzący dane testowe	Skrypt zawierający instrukcje <b>INSERT</b> , tworzące dane testowe.	Wymagane
Dane testowe w bazie	Sensowne (choć raczej nieliczne) dane testowe umieszczone w bazie.	Wymagane

Uwaga: dla danych typu data (i czas) użyć jawnych funkcji konwersji (np. **TO\_DATE**), a nie formatu domyślnego. Pamiętać o instrukcji **COMMIT**!

#### 5.3.2 Testy struktury danych – wymagania

Celem jest przetestowanie zaprojektowanej struktury danych, a w szczególności ograniczeń integralności. Należy wypróbować struktury przez zadawanie zapytań i próby naruszania więzów.

Nazwa	Opis	Ważność
Zapytania testowe	Dające się wykonać na danych testowych sensowne zapytania <b>SELECT</b> (kilka), nawiązujące do treści zadania.	Wymagane
Testy ograniczeń	Zdania DML ( <b>INSERT</b> , <b>UPDATE</b> , <b>DELETE</b> ) testujące skuteczność wybranych więzów integralności.	Wymagane

**Poprawianie utworzonych struktur** Jeśli utworzone już w bazie struktury danych wymagają poprawienia, należy je usunąć z bazy danych, poprawić ich definicje i wygenerować je ponownie.

### 5.4 Odbiór etapu 3

**Prowadzący odbiór** Wyniki etapu 3 sprawdzają mgr inż. Klara Borowa i dr inż. Szymon Kijas.

**Treść odbioru** Do odbioru należy przedstawić:

- diagram struktury danych w postaci pliku PDF, wygenerowany jak opisano poniżej.
- struktury utworzone w bazie danych, wraz z elementami uzupełniającymi oraz danymi testowymi (sprawdzenie odbędzie się wprost w bazie danych, nie trzeba przysyłać skryptów);
- wyniki testów, tj. pliki .LIS, jak opisano niżej.

**Przesłanie wyników etapu 3** Po zakończeniu tworzenia struktur danych, a przed wysłaniem ich do oceny, należy sprawdzić jego poprawność, analizując krytycznie wygenerowany skrypt SQL DDL i utworzone struktury w bazie danych. Następnie należy uporządkować formę graficzną diagramu i przygotować pliki do wysyłki.

1. Krytycznie obejrzyć wyniki etapu 3 w bazie danych. Za pomocą programu *Oracle SQL\*Developer* obejrzyć utworzone w bazie struktury. Zwrócić szczególną uwagę na poprawność nazw, typów danych, więzów integralności oraz indeksów.  
Uwaga: sprawdzający projekt będzie oceniać struktury oglądając je bezpośrednio w bazie danych.
2. Diagram zaopatrzyć w dane twórcy i zadania: na diagramie, w jednym z górnych rogów, umieścić jako notatkę dane twórcy i zadania: numer konta Oracle w formacie **BD2C###**, gdzie **###** jest numerem konta użytkownika w bazie Oracle, numer zadania, nazwisko i imię oraz adres e-mail autora (w miarę możliwości proszę używać adresów zawierających nazwisko nadawcy). Diagramy bez takiego opisu będą odrzucane!
3. Uporządkować formę graficzną diagramu modelu relacyjnego. Dopilnować, by wszystkie napisy były widoczne. Przesunąć elementy diagramu jak najbliżej górnej i lewej krawędzi, a następnie ustawić je tak, by proporcje rysunku przypominały proporcje pionowo ustawionej kartki papieru formatu A4. Notatkę z danymi autora i zadania pozostawić w lewym górnym rogu.
4. Wygenerować plik z diagramem modelu relacyjnego: za pomocą opcji *File / Print Diagram / To Image File* zapisać diagram w formacie PDF. Obejrzyć wygenerowany rysunek, w szczególności sprawdzić czy nie ma on zbyt dużych marginesów przy górnej i lewej krawędzi.  
Uwaga: nie należy tworzyć pliku z grafiką bitmapową i osadzać go w PDF. Akceptowane będą wyłącznie oryginalne wektorowe pliki PDF.
5. Plik z diagramem przekazać za pomocą systemu Studia jako sprawozdanie **Etap 3: diagram**.
6. Przygotować pliki .LIS. Sprawdzić czy w plikach tych widać zarówno instrukcje SQL jak i odpowiedzi bazy danych!
7. Pliki .LIS spakować do formatu .ZIP i przekazać za pomocą systemu Studia jako sprawozdanie **Etap 3: pliki**.

**Uwagi:**

- Prowadzący będzie oceniał struktury utworzone w bazie danych. Należy więc pozostawić tam odpowiednią zawartość wraz z danymi testowymi.
- Prace nie zawierające wymaganych informacji będą odrzucane.
- Wszelkie pytania proszę przysyłać e-mailem do osób sprawdzających etap 3.
- Pliki .LIS należy pakować wyłącznie do formatu .ZIP. Inne formaty, np. 7Z, nie będą akceptowane.

## 5.5 Ocena i poprawianie etapu 3

Z etapu 3 można otrzymać do 20 punktów: 15 z kategorii S – końcowa struktura danych, 5 z kategorii D – elementy dodatkowe (dane testowe, zapytania itd.). Na ocenę tę istotnie wpływa poprawność uwzględnienia uwag prowadzącego z etapu 2.

Wyniki sprawdzenia będą zapisane jako adnotacje na diagramie i przekazane przez mechanizm odsyłania sprawdzonego sprawozdania **Etap 3: diagram** w systemie Studia.

Jeśli pracę opatrzono adnotacją „Do poprawki” lub podobną, obowiązuje jej poprawienie. Przysługuje jedna poprawka, przy czym prawo do poprawki może być uzależnione od terminowego przekazania pierwszego „podejścia” do etapu.

**Forma odbioru poprawek** Poprawkę należy przesłać w wyznaczonym terminie. Plik z diagramem i pliki .LIS przekazać – jak poprzednio – za pomocą systemu Studia jako sprawozdania **Etap 3: diagram-poprawka** i **Etap 3: pliki-poprawka**.

## 5.6 Najczęstsze błędy w etapie 3

Poniższa tabela zestawia najczęstsze błędy popełniane w etapie 3. Podane tu kody błędów będą używane podczas sprawdzania prac.

Kod	Opis
IGNOR	Zignorowanie ważnych uwag ze sprawdzenia etapu 2 $\Rightarrow$ ocena 0
RESZT	Resztki – pozostawienie zbędnych, śmieciowych artefaktów w bazie
K-NSPEC	Kolumna PK nazywająca się ID (tylko)
K-NAZ	Nazwa jako PK
K-BRAK	Brak PK
K-NADM	PK błędny formalnie lub nadmiarowy (np. FK + sztuczny ID)
K-NM	Błędny PK w tabeli intersekcji
K-REKUR	FK rekurencyjny obowiązkowy
K-ZLOZ	Źle zrealizowany FK złożony
K-KASK	Zbędna lub brakująca kaskada kasowania
N-NAZ	Formalnie zła nazwa tabeli/kolumny: ze spacjami, znakami narodowymi itp.
N-TERM	Użycie błędnej terminologii (relacja zamiast klucz obcy, encja zamiast tabela itp.)
N-POWT	Zła konwencja nazw: zbędne prefiksy, zbędne powtórzenie nazwy tabeli, brak potrzebnego powtórzenia nazwy tabeli itp.
N-NM	Zła nazwa tabeli intersekcji (sklejenie nazw tabel)
N-SMIEC	Śmieci w nazwach więzów: zbędne numerki, spacje itp.
N-LITER	Literówka w nazwie
N-ORT	Błąd ortograficzny w nazwie lub opisie
N-ZLA	Niefortunna nazwa
N-SKROT	Skrót w nazwie
O-TAB	Brak komentarzy do tabel lub zbyt skąpy opis
O-KOL	Brak komentarzy do kolumn lub zbyt skąpe opisy
T-INT	Złe odwzorowanie typów, np. INTEGER, NUMBER bez parametrów
T-TYP	Niedopasowany typ danych lub zła długość
T-KWOT	Kwoty pieniężne, kursy walut, ceny itp. wyrażone zmiennoprzecinkowo

...

Kod	Opis
B-BRAK	Brak danych testowych
S-11	Związek 1-1
S-NM	Zła realizacja rozwinięcia związku wiele do wielu
S-PODT	Zakazana realizacja podtypów (związki 1-1 i łuk)
S-INF	Kolumna typu złożonego
S-NF	Złamanie NF
S-REDUND	Inna, mniej szkodliwa redundancja
W-ZLY	Źle napisany wyzwalacz
W-DEKL	Wyzwalacz realizujący ograniczenie wykonalne deklaratywnie
W-FKNT	Błędne (niepoprawione) wyzwalacze FKNT...
I-FK	Brak indeksu na FK
I-PK	Zła kolejność w PK
I-ZLOZ	Źle zaindeksowany klucz złożony (osobne indeksy zamiast indeksu złożonego)
I-REDUND	Zbędny indeks na FK (pokrywający się z początkiem klucza innego indeksu)
I-NADM	Zbędny indeks „do wyszukiwania”

## 6 Instrukcje

### 6.1 Tworzenie modelu związków encji – instrukcja

#### Utworzenie diagramu ER

1. Wywołać narzędzie Oracle SQL Developer Data Modeler. Jeśli w oknie programu nie pojawi się tzw. *browser*, czyli drzewo pokazujące składniki projektu, wybrać z menu **View / Browser**. Jeśli w prawej górnej części okna nie pojawi się obszar roboczy narzędzia do tworzenia diagramów ER, kliknąć prawym przyciskiem myszki na pozycji **Logical** w browserze i wybrać z lokalnego menu pozycję **Show**.
2. Zapisać nowy projekt: wybrać **File / Save** i zapisać nowy projekt pod nazwą **BD2C###**. Powstanie plik **BD2C###.dmd** oraz podkatalog **BD2C###** zawierający treść projektu.
3. Opisać diagram: wybrać z paska narzędzi narzędzie do rysowania notatek (**New note**) i w lewym górnym rogu obszaru roboczego umieścić notatkę, o treści: **ERD BD2C###, zadanie ##**, gdzie znaki **##...** należy zastąpić odpowiednimi numerami, w notatce tej należy też umieścić imię i nazwisko oraz adres e-mail autora.
4. Zapisać diagram: używając opcji **File / Save** zapisywać projekt co jakiś czas. Należy pamiętać o samodzielnym wykonywaniu kopii rezerwowych plików projektu!

**Tworzenie modelu ER** Czynności opisane w tej części należy wykonywać tak, by spełnić wymagania danego etapu.

1. Tworzenie encji: użyć przycisku **Entity** z paska narzędzi; narysować prostokąt w odpowiednim miejscu diagramu; podać nazwę encji, w polu **Short Name** podać kilkuleterowy skrót nazwy (bez spacji), zaś w polu **Preferred Abbreviation** wpisać nazwę w liczbie mnogiej i bez polskich znaków (będzie ona użyta do nazwania tabeli relacyjnej powstałej na podstawie encji).
2. Tworzenie związków: użyć odpowiedniego przycisku związku z paska narzędzi. Kliknąć w symbol encji najpierw po stronie „jeden”, potem po stronie „wiele” związku. W dialogu

opisu związku wpisać nazwy związku i ustalić stopień oraz opcjonalność związku. Kliknąć na obszarze roboczym prawym przyciskiem myszki i włączyć opcję **Show / Labels**, by uwidocznić nazwy na diagramie.

3. „Prostowanie” związków: kliknąć na obszarze roboczym prawym przyciskiem myszki i wybrać opcję **Straighten Lines**.
4. Uszczegółowienie encji: dwukrotnie kliknąć w symbol encji. Użyć wywołanego dialogu do edycji właściwości encji i atrybutów oraz unikalnych identyfikatorów, zgodnie z wymaganiami podanymi powyżej. Dla atrybutów podać przybliżone typy danych i właściwe długości (użyć **Datatype / Logical**).
5. Tworzenie podtypów (opcjonalne): w celu stworzenia podtypu utworzyć nową encję poza nadtypem, a następnie w jej właściwościach w polu **Super Type** wybrać nazwę odpowiedniej encji-nadtypu.
6. Dokumentacja: opisy dla encji i atrybutów należy wpisywać w pola **Comments in RDBMS**. Wpisane tam teksty będą w przyszłości zapisane do bazy danych za pomocą wygenerowanych instrukcji **SQL COMMENT ON . . .** W opisach nie należy zatem używać słów *encja* czy *atrybut*, gdyż docelowo będą to opisy tabel i kolumn.

### Zapis diagramu ERD do pliku PDF

1. Upewnić się czy diagram spełnia wymagania, także co do sposobu formatowania oraz wymaganego opisu (dane zadania, dane i adres e-mail twórcy).
2. Jeśli symbole encji mają włączone wypełnienie kolorem, to je wyłączyć klikając prawym przyciskiem myszki na symbolu encji i wybierając **Format**.
3. Jeśli było włączone wyświetlanie siatki, to je wyłączyć klikając prawym przyciskiem myszki na tle diagramu i wybierając **Show / Grid**.
4. Kliknąć prawym przyciskiem myszki na tle diagramu i wybrać **Print Diagram / To PDF File**.

Uwaga: nie należy tworzyć pliku z grafiką bitmapową i osadzać go w PDF, np. używając Worda. Akceptowane będą wyłącznie oryginalne wektorowe pliki PDF.

## 6.2 Projekt struktur danych – instrukcja

Czynności projektowe – o ile nie napisano inaczej – wykonywane mają być za pomocą narzędzia *Data Modeler*.

### Utworzenie modelu relacyjnego

1. **Sprawdzenia:** – czy opisy dokumentacyjne do encji i atrybutów znajdują się we właściwościach **Comment in RDBMS**; – czy encje mają nazwy w liczbie mnogiej wpisane do **Preferred Abbreviation**.
2. Utworzenie modelu relacyjnego: za pomocą opcji **Design/Engineer to Relational Model** utworzyć wstępną wersję modelu tabel.  
Uwaga: przed wywołaniem operacji **Engineer** przejść na zakładkę **General Options** i wyłączyć opcję **Apply template for FK columns**, po czym wywołać **Apply selection**. Po wygenerowaniu struktury relacyjnej upewnić się, że nazwy kolumn kluczy obcych nie są prefiksowne nazwą tabeli.
3. Opisanie diagramu: przejść do diagramu modelu relacyjnego. Wybrać z paska narzędzi narzędzie do rysowania notatek (**New note**) i w lewym górnym rogu obszaru roboczego umieścić notatkę, o treści: **BD2C###**, **zadanie ##** gdzie znaki **##**... należy zastąpić odpowiednimi numerami. W notatce tej należy też umieścić imię i nazwisko oraz adres e-mail autora.



4. Zapisywanie projektu: używając opcji File / Save zapisywać projekt co jakiś czas. Należy pamiętać o samodzielnym wykonywaniu kopii rezerwowych plików projektu!

### Poprawianie i uzupełnianie projektu struktur relacyjnych

1. Edycja właściwości kolumn: podwójne kliknięcie na nazwie kolumny w drzewie *browsera* powoduje otwarcie dialogu właściwości kolumny. Można tam zdefiniować m.in. wartość domyślną oraz listę wartości dopuszczalnych.
2. Dodawanie indeksów: używając dialogu właściwości tabeli należy dodać potrzebne indeksy dla kluczy obcych. Nie projektować osobno indeksów dla kluczy głównych i unikalnych. Dopilnować, by indeksy się nie dublowały, tzn. by żaden z kluczy indeksowania (włączając PK i UK!) nie był równy początkowej części innego.
3. Przenaszalność związków i kaskada kasowania: jeśli któryś ze związków powinien być nie-przenaszalny, należy wyłączyć właściwość *Transferable* odpowiedniego klucza obcego. W celu utworzenia klucza obcego z klauzulą ON DELETE CASCADE należy zmienić właściwość *Delete Rule* klucza na *CASCADE*.

### Tworzenie „modelu fizycznego” – instrukcja

1. Utworzyć „model fizyczny”: na węźle **Physical Model** drzewa *browsera* kliknąć prawym przyciskiem myszki i wybrać *New*, z listy wybrać *Oracle Database* (najnowszą wersję).  
Uwaga: utworzony model otwiera się klikając na jego nazwie prawym uchem myszki i wybierając *Open*.)
2. Utworzyć definicję użytkownika „końcowego”: w węźle **Physical Model** wybrać pozycję *Users* i utworzyć definicję użytkownika o nazwie **BD2C###\_APP**.
3. Przyznawanie praw dostępu do tabel: w „modelu fizycznym” otworzyć dialog właściwości dla każdej z tabel i używając przycisku *Permissions* utworzyć prawa dostępu dla użytkownika **BD2C###\_APP**: *Select*, *Insert*, *Update* i *Delete*.
4. Ustawienie możliwości odroczenia walidacji więzów: w „modelu fizycznym” dla każdej z tabel wybrać po kolei klucze obce i dla każdego z nich ustawić właściwość *Deferrable* na *Yes*.

## 6.3 Generowanie struktur danych – instrukcja

Na podstawie projektu należy wygenerować skrypt tworzący obiekty. Skrypt ten należy przejrzeć! Jeśli stwierdzi się w nim usterki, należy poprawić projekt struktur i ponowić generację. Następnie należy uruchomić skrypt i utworzyć struktury w bazie danych.

### Generowanie i wykonanie skryptu SQL DDL

1. Generowanie skryptu DDL: za pomocą opcji *Generate DDL* narzędzia *Data Modeler* (przycisk z symbolem walca, na pasku narzędzi po prawej stronie) wygenerować skrypt DDL; sprawdzić w zakładce *Tables* czy wybrane są wszystkie zaprojektowane tabele. Skrypt starannie przejrzeć. Jeśli jest OK, używając przycisku *Save* skrypt zapisać.
2. Wykonanie skryptu: używając programu *SQL Developer* wykonać jako użytkownik **BD2C###** wygenerowany uprzednio skrypt. Instrukcja tworzenia użytkownika powinna spowodować błąd (użytkownik jest już utworzony przez administratora), pozostałe instrukcje powinny się wykonać bezbłędnie. W razie wystąpienia błędów usunąć (DROP) utworzone obiekty, dokonać poprawek w projekcie i powtórzyć kolejne czynności.

## Dane testowe – instrukcja

1. Napisać skrypt tworzący dane testowe: za pomocą dowolnego edytora tekstowego utworzyć skrypt, zawierający zdania INSERT tworzące dane testowe. Skrypt ten wykonać za pomocą programu SQL Developer jako użytkownik BD2C###.
2. Nie zapomnieć o instrukcji COMMIT!

**Elementy uzupełniające strukturę** Elementy uzupełniające należy zaprojektować w narzędziu *Data Modeler*, a następnie wygenerować skrypty tworzące je w bazie danych.

1. Otworzyć „model fizyczny”: kliknąć prawym przyciskiem myszki w węzeł *Physical Models* i wybrać *Open*.
2. Utworzyć definicję użytkownika – właściciela obiektów: wybrać węzeł *Users* i utworzyć definicję użytkownika o nazwie BD2C###.
3. Utworzyć definicję użytkownika „końcowego”: wybrać węzeł *Users* i utworzyć definicję użytkownika o nazwie BD2C###\_APP.
4. Zdefiniować synonimy dla użytkownika nieuprzywilejowanego: w węźle *Synonyms* utworzyć synonimy dla wszystkich tabel (i ewentualnie perspektyw) projektu. Synonim nazwać tak samo, jak nazywa się wskazywany obiekt, w polu *User* wybrać użytkownika BD2C###\_APP, w polu *Object Owner* wpisać BD2C###, zaś w polu *Public* pozostawić wartość *No*.
5. Wygenerować skrypt tworzący elementy uzupełniające: za pomocą opcji *Generate DDL* narzędzia *Data Modeler* (przycisk z symbolem walca, na pasku narzędzi po prawej stronie) wygenerować skrypt DDL: po wciśnięciu przycisku *Generate* zgasić zaznaczenie na najwyższym poziomie drzewa (przy *Oracle Database...*). Następnie na dole dialogu wybrać zakładkę *Synonyms* i na niej zaznaczyć wszystkie stworzone synonimy. Wygenerować skrypt, a następnie starannie go przejrzeć. Jeśli jest OK, używając przycisku *Save* skrypt zapisać.  
Uwaga: instrukcje w skrypcie powinny wyglądać tak: `CREATE OR REPLACE SYNONYM xxx FOR BD2C###.xxx`; Jeśli wyglądają inaczej, należy je przerobić zęcznie”.
6. Wykonać skrypt: używając programu SQL Developer wykonać jako użytkownik BD2C###\_APP wygenerowany skrypt z synonimami.  
Uwaga: pamiętać o tym, że użytkownik ten ma osobne hasło, jeśli więc go nie zmieniono, to jest ono takie, jak hasło początkowe użytkownika BD2C###!
7. Zaprojektować inne rodzaje elementów uzupełniających struktury danych: w odpowiednich węzłach „modelu fizycznego” utworzyć i odpowiednio zdefiniować sekwencje, wyzwalacze itp.
8. Wygenerować skrypt tworzący elementy uzupełniające: za pomocą opcji *Generate DDL* narzędzia *Data Modeler* (przycisk z symbolem walca, na pasku narzędzi po prawej stronie) wygenerować skrypt DDL: po wciśnięciu przycisku *Generate* zgasić zaznaczenie na najwyższym poziomie drzewa (przy *Oracle Database...*). Następnie na dole dialogu wybrać zakładkę *Synonyms* i na niej sprawdzić czy wyłączone jest zaznaczenie wszystkich synonimów. Wybrać odpowiednie zakładki dla innych typów utworzonych elementów uzupełniających i włączyć zaznaczenia. Wygenerować skrypt, a następnie starannie go przejrzeć. Jeśli jest OK, używając przycisku *Save* skrypt zapisać.
9. Wykonać skrypt: używając programu SQL Developer wykonać jako użytkownik BD2C### wygenerowany skrypt tworzący elementy uzupełniające. Wynik obejrzeć w bazie danych!

## Testy struktury danych – instrukcja

1. Przetestować dane testowe za pomocą zapytań: utworzyć skrypt, zawierający zdania `SELECT` testujące dane. Zdania te powinny nawiązywać do treści zadania. Na początku skryptu umieścić dyrektywy:

```
set echo on
set linesize 300
set pagesize 500
spool BD2C###.TEST.LIS
zaś na końcu dyrektywę:
spool off
```

Skrypt ten zapisać i wykonać za pomocą programu SQL Developer jako użytkownik `BD2C###`, korzystając w *Worksheet* z komendy `@nazwa_skryptu.sql`. Sprawdzić czy w pliku `.LIS` widać treść zdań SQL (powinny być widoczne na skutek dyrektywy `set echo on`) oraz wyniki tych zapytań.

2. Testować dostęp przez synonimy: skrypt testowy wykonać za pomocą programu SQL Developer jako użytkownik `BD2C###_APP` i zaobserwować wyniki. Działanie powinno być identyczne, jak dla użytkownika `BD2C###`, tj. wszystkie zapytania powinny się wykonać poprawnie.
3. Napisać zdania DML testujące ograniczenia: utworzyć skrypt, zawierający zdania DML testujące działanie wybranych ograniczeń. Na początku skryptu umieścić dyrektywę `spool BD2C###.DML.LIS`, zaś na końcu dyrektywę `spool off`. Skrypt ten wykonać za pomocą programu SQL Developer jako użytkownik `BD2C###`, podobnie jak poprzednio. Sprawdzić czy w pliku `.LIS` widać treść zdań SQL i wyniki ww. testów.