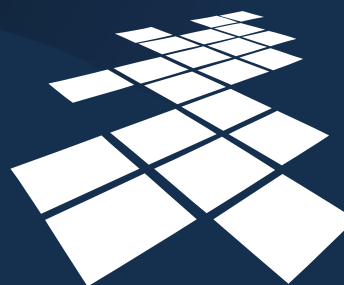


Ćwiczenie 7 - DDL

Relacje i ograniczenia
integralnościowe.



UCZELNIA
ONLINE

Ćwiczenie 7 – DDL

Podczas dotychczasowych ćwiczeń z Systemów Baz Danych wykonywaliście Państwo zadania przy wykorzystaniu relacji PRACOWNICY, ZESPOLY i ETATY. Poznaliście Państwo polecenia pozwalające na wykonywanie zapytań do relacji oraz wstawianie, modyfikację i usuwanie z nich danych. Celem siódmego ćwiczenia z Systemów Baz Danych jest zapoznanie Państwa z poleceniami pozwalającymi na tworzenie relacji, ich modyfikację i usuwanie. Poznacie Państwo również typy danych, jakie mogą być składowane w bazie danych

Wymagania:

Do wykonania ćwiczenia konieczna jest znajomość tematyki omawianej na poprzednich ćwiczeniach z SBD, w szczególności z wykonywania zapytań, oraz wstawiania, modyfikacji i usuwania danych z relacji.



Plan ćwiczenia

- Wprowadzenie do laboratorium.
- Tworzenie relacji.
- Typy danych.
- Ograniczenia integralnościowe (klucz główny, wartość unikalna, obowiązkowa wartość atrybutu, ograniczenie domeny atrybutu, klucz obcy).
- Tworzenie relacji przez podzapytanie.

Ćwiczenie 7 - DDL (2)

Ćwiczenie rozpoczniemy od wprowadzenia do laboratorium. Następnie przedstawimy polecenie `CREATE TABLE` pozwalające na tworzenie relacji w bazie danych i przedstawimy typy danych, jakie można nadać atrybutom relacji. Na kolejnym etapie ćwiczenia poznacie państwo koncepcję ograniczeń integralnościowych oraz składnię poleceń pozwalającą na ich definiowanie. Ostatnim tematem dotyczącym tworzenia relacji jest automatyczne wypełnianie relacji danymi podczas ich tworzenia.



Plan ćwiczenia – cd.

- Modyfikowanie schematu relacji.
- Zarządzanie ograniczeniami integralnościowymi.
- Zmiana nazwy i usuwanie relacji.
- Zadania
- Podsumowanie.

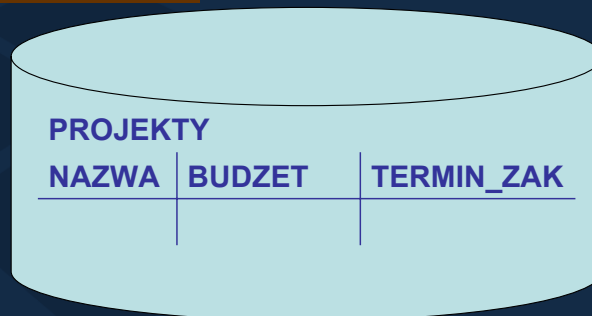
Ćwiczenie 7 - DDL (3)

Następnie przejdziemy do omówienia poleceń pozwalających na: modyfikowanie istniejących relacji oraz ograniczeń integralnościowych, oraz na zmianę nazwy relacji i ich usuwanie. Każdy z tematów omawianych na ćwiczeniu zakończony jest zadaniami do samodzielnego wykonania. Na końcu ćwiczenia przedstawimy państwu kilka dodatkowych zadań, które powinniście państwo wykonać w celu nabrania wprawy w posługiwaniu się poleceniami przedstawionymi na ćwiczeniu. Ćwiczenie zakończymy slajdem podsumującym omówioną tematykę.



Wprowadzenie do laboratorium

```
CREATE TABLE projekty (  
  nazwa CHARACTER VARYING (100),  
  budżet NUMERIC(10,2),  
  termin_zak DATE  
);
```



Ćwiczenie 7 - DDL (4)

Podczas dotychczasowych ćwiczeń z Systemów Baz Danych wykonywaliście Państwo zadania przy wykorzystaniu relacji PRACOWNICY, ZESPOLY i ETATY. Poznaliście Państwo polecenia pozwalające na wykonywanie zapytań do relacji oraz wstawianie, modyfikację i usuwanie z nich danych. Wiedza ta jednak nie jest jednak specjalnie przydatna, jeżeli nie można samemu stworzyć własnych relacji. Na tym ćwiczeniu poznacie Państwo polecenia pozwalające na tworzenie relacji (przykład na slajdzie) oraz ich modyfikację i usuwanie. Poznacie Państwo również koncepcję ograniczeń integralnościowych, które pozwalają na utrzymywanie spójności w bazie danych.



Tworzenie relacji

1 CREATE TABLE nazwa_relacji

```
(nazwa_atrybutu typ (rozmiar) [DEFAULT wartość_domyślna],
nazwa_atrybutu typ (rozmiar) [DEFAULT wartość_domyślna],
....
);
```

2 CREATE TABLE projekty (

```
nazwa CHARACTER VARYING (100),
budżet NUMERIC(10,2) DEFAULT 200000,
termin_zak DATE DEFAULT SYSDATE +
INTERVAL '5-6' YEAR (1) TO MONTH
);
```

Ćwiczenie 7 - DDL (5)

Nowe relacje tworzy się za pomocą polecenia CREATE TABLE, za którym podaje się nazwę relacji i listę atrybutów tej relacji. Listę atrybutów tworzy się wymieniając po przecinku wyrażenia złożone z nazwy atrybutu, jego typu i rozmiaru typu oraz opcjonalnie domyślnej wartości atrybutu. Domyślna wartość jest wstawiana do relacji w sytuacji, gdy nie podano w poleceniu INSERT wartości dla wszystkich atrybutów. Na slajdzie pokazano ogólną składnię polecenia CREATE TABLE (1). Przykład (2) zawiera polecenie tworzące relację PROJEKTY zawierającą atrybuty NAZWA, BUDZET i TERMIN_ZAK. Atrybut NAZWA reprezentuje nazwę projektu i jest typu łańcuchowego o maksymalnym rozmiarze 100 (typy zostaną dokładnie opisane na następnych slajdach). Atrybut BUDZET reprezentuje budżet projektu i jest typu liczbowego. Domyślną wartością tego atrybutu jest 200000. Atrybut TERMIN_ZAK reprezentuje termin zakończenia projektu, a domyślną wartością tego atrybutu jest wartość wyrażenia SYSDATE + INTERVAL '5-6' YEAR (1) TO MONTH, które oznacza datę systemową w momencie wstawiania krotki powiększoną o 5 lat i 6 miesięcy. Przykładowe polecenia wykorzystujące wartości domyślne dla atrybutów wyglądają następująco:

```
INSERT INTO projekty(nazwa) VALUES ('System informatyczny dziekanatu');
```

Polecenie to spowoduje wstawienie do relacji PROJEKTY projektu o nazwie „System informatyczny dziekanatu”, o budżecie 200000 zł i terminie zakończenia za pięć i pół roku od daty wstawienia krotki.

```
INSERT INTO projekty(nazwa, budżet) VALUES ('System informatyczny dziekanatu', 100000);
```

Polecenie to spowoduje wstawienie do relacji PROJEKTY projektu o nazwie „System informatyczny dziekanatu”, o budżecie 100000 zł (podana wartość „nadpisuje” wartość domyślną) i terminie zakończenia za pięć i pół roku od daty wstawienia krotki.

Należy tutaj zwrócić uwagę na jeszcze jeden fakt. Otóż w niektórych SZBD wykonanie dowolnego polecenia DDL (w tym i CREATE TABLE) wiąże się z niejawnym wykonaniem zatwierdzenia wszystkich zmian (COMMIT). W konsekwencji nie jest możliwe wycofanie poleceń DDL. Jest tak np. w SZBD Oracle, natomiast w DB2 możliwe jest wycofywanie np. polecenia tworzącego relację.



Typy danych – łańcuchy

Typ	Opis
CHARACTER(n) CHAR(n)	Typ łańcuchowy o stałej długości.
CHARACTER VARYING(n) CHAR VARYING(n)	Typ łańcuchowy o zmiennej długości.
NATIONAL CHARACTER(n) NATIONAL CHAR(n)	Typ łańcuchowy o stałej długości, o predefiniowanym narodowym zbiorze znaków.
NATIONAL CHARACTER VARYING (n) NATIONAL CHAR VARYING(n) NCHAR VARYING(n)	Typ łańcuchowy o zmiennej długości, o predefiniowanym narodowym zbiorze znaków.

Ćwiczenie 7 - DDL (7)

Istnieje kilka typów danych pozwalających na definiowanie atrybutów składających tekst. Typ CHARACTER (w skrócie CHAR) reprezentuje łańcuchy znaków o długości równej zdefiniowanej (n). Jeżeli wstawiony łańcuch jest krótszy, jest on uzupełniany znakami spacji do odpowiedniej długości. Według FIPS (*Federal Information Processing Standards*) powinna istnieć możliwość definiowania łańcuchów o długości 1000 znaków, ale maksymalna wartość tutaj zależy od producenta SZBD.

Typ CHARACTER VARYING (w skrócie CHAR VARYING) jest typem podobnym do typu CHAR. Tutaj jednak podaje się nie długość składowanego łańcucha, a jego maksymalną długość. Możliwe jest zatem składowanie łańcuchów krótszych niż podany rozmiar i nie są one uzupełniane spacjami. Podobnie jak w przypadku typu CHARACTER powinna istnieć możliwość definiowania łańcuchów o maksymalnej długości równej 1000 znaków, ale faktyczna maksymalna wartość tutaj zależy od producenta SZBD.

Typy CHARACTER i CHARACTER VARYING posiadają również swoją wersję, która pozwala na przechowywanie łańcuchów zawierający znaki narodowe z predefiniowanego zestawu znaków. Te dodatkowe wersje typów łańcuchowych nazywają się prawie tak samo, jak wersje oryginalne. Jediną różnicą jest to, że ich nazwa jest uzupełniona o słowo NATIONAL.

Parametr (n), oznaczający długość albo maksymalną długość łańcucha, dla każdego z typów przedstawionych na slajdzie jest opcjonalny i domyślnie wynosi 1.



Typy danych – liczby

Typ	Opis
INTEGER, INT SMALLINT	Typ całkowitoliczbowy
NUMERIC(p,s) DECIMAL(p,s)	Typ liczbowy o definiowanej precyzji i skali
FLOAT(b) DOUBLE PRECISION REAL	Typ zmiennoprzecinkowy

Ćwiczenie 7 - DDL (8)

Typy INTEGER, INT i SMALLINT pozwalają na przechowywanie wartości całkowitych ze znakiem. Według standardu SQL-99 typy INTEGER i INT są identyczne. Według FIPS powinny pozwalać na przechowywanie liczb złożonych przynajmniej z 9 cyfr. SMALLINT powinien mieć dokładność mniejszą lub równą dokładności typów INTEGER i INT. Według FIPS typ ten powinien pozwalać na przechowywanie liczb złożonych przynajmniej z 4 cyfr.

Typy NUMERIC i DECIMAL są typami liczbowymi stałoprzecinkowymi o możliwości definiowania precyzji (p) i skali (s). Precyzja określa ile cyfr liczby ma być przechowywanych, a skala ile z tych cyfr ma stanowić część ułamkową rozwinięcia dziesiętnego składowanej liczby. Precyzja musi być liczbą całkowitą większą od zera. Według FIPS powinna istnieć możliwość podawania wartości przynajmniej do 15, ale istnieją SZBD pozwalające podać tutaj nawet 38. Skala musi być całkowitą, dodatnią liczbą mniejszą od precyzji. Zarówno precyzja, jak i skala są opcjonalne i można je pominąć. W sytuacji, gdy chcemy podać skalę, należy podać również precyzję. Domyślną wartością skali jest 0, a domyślną wartością precyzji jest 1.

Typ FLOAT jest typem zmiennoprzecinkowym o definiowanej precyzji (b). Przez precyzję rozumie się tutaj maksymalną liczbę bitów, jaką można przeznaczyć na mantysę. Typy REAL i DOUBLE PRECISION są liczbami ze zdefiniowaną precyzją. Standard nie określa tutaj wartości precyzji, ale wymaga, aby precyzja REAL była mniejsza od DOUBLE PRECISION.



Typy danych – daty i czas

Typ	Opis
DATE	Reprezentuje daty.
TIME(n)	Reprezentuje czas z dokładnością do części ułamkowych sekundy
TIMESTAMP(n)	Data i czas (połączenie DATE i TIME)
INTERVAL specyfikacja	Reprezentuje okres czasu

Typ DATE pozwala na definiowanie atrybutów przechowujących daty. Poprawną datą jest tutaj dowolna data z kalendarza gregoriańskiego pomiędzy 01 stycznia 1 roku a 31 grudnia 9999. W SZBD Oracle typ DATE pozwala również na przechowywanie czasu w postaci wartości ułamkowych (1=dzień, 0,5=12 godzin itp.).

Typ TIME pozwala przechowywać informacje o czasie dnia z dokładnością do ułamkowych części sekundy. Parametr określa liczbę pozycji dziesiętnych rozwinięcia dziesiętnej części sekundy. Musi istnieć możliwość podania tutaj przynajmniej wartości 6. Parametr (n) jest opcjonalny, a jego domyślna wartość wynosi 0.

Typ TIMESTAMP reprezentuje połączenie możliwości typów DATE i TIME i pozwala na składowanie zarówno dat, jak i czasu. Podobnie jak typy DATE i TIME pozwala na przechowywanie w atrybutach dat z kalendarza gregoriańskiego pomiędzy 01 stycznia 1 roku a 31 grudnia 9999 oraz czasu z dokładnością do ułamkowych części sekundy. Parametr (n) ma identyczne znaczenie jak w typie TIME i również jest opcjonalny.

Typ INTERVAL reprezentuje okres czasu. Okres czasu definiowany jest za pomocą pewnej liczby różnych „jednostek” takich jak: lata, miesiące, dni, godziny itp. Na nazwę typu składa się słowo kluczowe INTERVAL i specyfikacja przechowywanych wielkości. Specyfikacja ma postać:

najdłuższy_okres [TO najkrótszy okres]

gdzie: najdłuższy_okres może przyjmować następujące wartości:

YEAR(n), MONTH(n), DAY(n), HOUR(n), MINUTE(n), SECOND(n,f) (n to liczba cyfr liczby danych jednostek czasu, jaka może być składowana, a f to liczba cyfr rozwinięcia dziesiętnej części sekundy),

A najkrótszy_okres może przyjmować wartości:

MONTH, DAY, HOUR, MINUTE, SECOND(f).

Podając opcjonalny najkrótszy okres należy pamiętać o tym, że musi on być krótszy od najdłuższego okresu. Przykładowo, poprawnym typem jest INTERVAL YEAR TO MONTH, ale nie jest nim INTERVAL MONTH TO YEAR. Nie jest również możliwe łączenie w jednej specyfikacji roku (YEAR) i miesiąca (MONTH) z pozostałymi okresami. Tym samym, lista poprawnych typów wygląda następująco:

- INTERVAL YEAR – pozwala na przechowywanie pewnej liczby lat.
- INTERVAL MONTH – pozwala na przechowywanie pewnej liczby miesięcy.
- INTERVAL YEAR TO MONTH – pozwala na przechowywanie pewnej liczby lat i miesięcy.
- INTERVAL DAY – pozwala na przechowywanie pewnej liczby dni.
- INTERVAL HOUR – pozwala na przechowywanie pewnej liczby godzin.
- INTERVAL MINUTE – pozwala na przechowywanie pewnej liczby minut.
- INTERVAL SECOND – pozwala na przechowywanie pewnej liczby sekund.
- INTERVAL DAY TO HOUR – pozwala na przechowywanie pewnej liczby dni i godzin.
- INTERVAL DAY TO MINUTE – pozwala na przechowywanie pewnej liczby dni, godzin i minut.
- INTERVAL DAY TO SECOND – pozwala na przechowywanie pewnej liczby dni, godzin, minut i sekund.
- INTERVAL HOUR TO MINUTE – pozwala na przechowywanie pewnej liczby godzin i minut.
- INTERVAL HOUR TO SECOND – pozwala na przechowywanie pewnej liczby godzin, minut i sekund.
- INTERVAL MINUTE TO SECOND – pozwala na przechowywanie pewnej liczby minut i sekund.

Do powyższych typów, zgodnie z wcześniejszym opisem można również dodać parametry określające dokładność i maksymalną wielkość przechowywanych wartości.

W SZBD Oracle możliwe jest definiowanie relacji z atrybutami jedynie typów: INTERVAL YEAR TO MONTH i INTERVAL DAY TO SECOND. Nie stanowi to jednak dużego problemu, gdyż są to typy najbardziej ogólne.



Typy danych – inne

Typ	Opis
BLOB	Binarne dane o dużych rozmiarach.
CLOB	Dane znakowe o dużych rozmiarach.

Typ BLOB służy do przechowywania danych binarnych o dużych rozmiarach, np. plików graficznych, filmów, plików dźwiękowych itp.

Typ CLOB służy do przechowywania danych znakowych o dużych rozmiarach (większych niż pozwalają na to typy takie jak CHARACTER VARYING), np. stron HTML, plików XML, książek itp.



Zadanie (1)

- Utwórz relację ZWIERZETA o następujących atrybutach:

Nazwa	Typ
GATUNEK	Łańcuchowy o zmiennej długości (max 100 znaków).
JAJORODNY	Łańcuchowy o stałej długości (1 znak) (przechowuje wartości T/N).
LICZBA_KONCZYN	Liczbowy, (max 2 cyfry).
DATA_ODKRYCIA	Typ reprezentujący datę.



Rozwiązanie (1)

```
CREATE TABLE zwierzeta (  
  gatunek CHARACTER VARYING (100),  
  jajorodny CHAR(1),  
  liczba_konczyn NUMERIC(2),  
  data_odkrycia DATE  
);
```

Slajd pokazuje rozwiązanie zadania (1), którego treść przytoczono poniżej.

Utwórz relację ZWIERZETA o następujących atrybutach:

GATUNEK:

Typ łańcuchowy o zmiennej długości (max 100 znaków).

JAJORODNY:

Typ łańcuchowy o stałej długości (1 znak) (przechowuje wartości T/N).

LICZBA_KONCZYN:

Typ liczbowy, (max 2 cyfry).

DATA_ODKRYCIA:

Typ reprezentujący datę.



Ograniczenia integralnościowe

1

```
CREATE TABLE nazwa_relacji
    (nazwa_atrybutu typ (rozmiar) [DEFAULT wartość_domyślna]
      [ [CONSTRAINT nazwa_ogr] ograniczenie_atr]...,
    nazwa_atrybutu typ (rozmiar) [DEFAULT wartość_domyślna]
      [ [CONSTRAINT nazwa_ogr] ograniczenie_atr]...,
    ....
    [ [CONSTRAINT nazwa_ogr] ograniczenie_rel, ...] );
```

2

```
CREATE TABLE projekty (
    nazwa CHARACTER VARYING (100) PRIMARY KEY,
    budżet NUMERIC(10,2),
    termin_zak DATE CONSTRAINT niepusty_termin NOT NULL,
    CHECK (budżet > 100000) );
```

Ćwiczenie 7 - DDL (14)

Dane składowane w relacjach opisują jakiś wybrany fragment rzeczywistości. Niestety, łatwo jest wprowadzić do relacji dane, które rzeczywistości nie mogą odpowiadać w żaden sposób. Przykładowo, można wprowadzić do bazy danych pracowników zatrudnionych tysiąc lat temu, albo samochód o ujemnej maksymalnej prędkości. W celu zapobiegania takim sytuacjom stosuje się tzw. ograniczenia integralnościowe. Jest to zbiór warunków, które muszą być spełnione przez dane składowane w bazie danych, aby można je było uznać za poprawne. Każde polecenie SQL, które spowodowałoby naruszenie warunków integralnościowych jest wycofywane. Slajd pokazuje rozszerzoną składnię polecenia CREATE TABLE, dzięki której można podać ograniczenia integralnościowe (1). Istnieją dwa rodzaje ograniczeń integralnościowych: ograniczenia atrybutu i ograniczenia relacji. Ograniczenia atrybutu dotyczą jedynie pojedynczego atrybutu relacji. Definiuje się je na liście atrybutów po podaniu wartości domyślnej atrybutu. Ograniczenia relacji dotyczą więcej niż jednego atrybutu relacji i definiuje się je jako osobny wpis na liście atrybutów. Definicja ograniczenia integralnościowego rozpoczyna się od opcjonalnego podania jego nazwy po słowie kluczowym CONSTRAINT, po którym podaje się właściwą definicję. Poszczególne typy ograniczeń wraz ze sposobem definiowania zostaną opisane na kolejnych slajdach. Przykładowe definicje ograniczeń integralnościowych można zobaczyć na przykładowym poleceniu tworzącym relację PROJEKTY (2). Tutaj, ograniczenia integralnościowe zostały założone na atrybutach NAZWA i TERMIN_ZAK. W drugim przypadku podano również nazwę ograniczenia (NIEPUSTY_TERMIN). Ograniczenia relacji dotyczą kilku atrybutów relacji i definiowane są w ramach listy atrybutów, za właściwymi atrybutami. Podobnie jak w przypadku ograniczeń atrybutów, definicję ograniczenia relacji rozpoczyna się od opcjonalnego podania jego nazwy, a następnie podania właściwej definicji.

Na przykładowym poleceniu (2) zdefiniowano ograniczenie integralnościowe relacji „CHECK (budżet > 100000)”. Ograniczenie to dotyczy tylko jednego atrybutu, ale mogłoby dotyczyć większej ich liczby.



Klucz podstawowy

```

1 CREATE TABLE projekty (
    nazwa CHARACTER VARYING (100)
    CONSTRAINT pk_proj PRIMARY KEY,
    budżet NUMERIC(10,2),
    termin_zak DATE
);

```

```

2 CREATE TABLE projekty (
    numer NUMERIC(6),
    nazwa CHARACTER VARYING (100),
    budżet NUMERIC(10,2),
    termin_zak DATE,
    CONSTRAINT pk_proj PRIMARY KEY (numer, nazwa)
);

```

Ćwiczenie 7 - DDL (16)

Jednym z najważniejszych ograniczeń integralnościowych jest tzw. klucz podstawowy. Żaden z atrybutów wchodzących w skład klucza podstawowego nie może przyjmować wartości NULL. Dodatkowo wymagane jest, aby żadna kombinacja wartości tych atrybutów nie wystąpiła więcej niż raz. Przykład (1) pokazuje sposób definicji ograniczenia „klucz podstawowy” jako ograniczenia atrybutu. Aby zdefiniować klucz podstawowy jako ograniczenie atrybutu wystarczy za typem atrybutu podać opcjonalną nazwę ograniczenia, a następnie słowa PRIMARY KEY. Klucz podstawowy zdefiniowany jako ograniczenie atrybutu dotyczy tylko jednego atrybutu. Dzięki ograniczeniu zdefiniowanemu w przykładzie (1), żadna NAZWA w relacji PROJEKTY nie może się powtórzyć. Przykład (2) pokazuje definicję ograniczenia „klucz podstawowy” jako ograniczenia relacji. Aby zdefiniować klucz podstawowy jako ograniczenie relacji, po opcjonalnej nazwie ograniczenia podaje się słowa PRIMARY KEY a następnie w nawiasie listę atrybutów wchodzących w skład klucza. Dzięki ograniczeniu zdefiniowanemu w przykładzie (2), żadna kombinacja wartości atrybutu NUMER i atrybutu NAZWA nie może się powtórzyć. Przykładowo, może wystąpić wiele krotek o wartości atrybutu NUMER równej 1, ale muszą one mieć różne NAZWY. Może również wystąpić wiele krotek i wartości atrybutu NAZWA równej „System informatyczny dla dziekanatu”, jednak wówczas muszą one mieć różne wartości na atrybucie NUMER.

Jak łatwo zauważyć, zbiór atrybutów podlegających ograniczeniu integralnościowemu „klucz podstawowy” jednoznacznie identyfikuje krotkę. Innymi słowy, znając wartości atrybutów wchodzących w skład klucza podstawowego, wiemy, że istnieje co najwyżej jedna krotka o takiej kombinacji wartości. W relacji PRACOWNICY, którą poznaliście państwo podczas poprzednich ćwiczeń, kluczem głównym jest atrybut ID_PRAC. Nie jest kluczem głównym np. atrybut NAZWISKO, a zatem możliwe jest wstawienie kilku pracowników, o takim samym nazwisku.

Uwaga! Relacja może posiadać co najwyżej jeden klucz podstawowy.



Zadanie (2)

- Utwórz relację KLIENCI o następujących atrybutach i ograniczeniach:

Nazwa	Typ
PESEL	Łańcuchowy o stałej długości (11 znaków). Klucz podstawowy (ograniczenie atrybutu).
ADRES	Łańcuchowy o zmiennej długości (max 100 znaków)
WIEK	Liczbowy, (max 2 cyfry).
WSPOLPRACA_OD	Typ reprezentujący datę.

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „klucz podstawowy”.



Rozwiązanie (2)

```
CREATE TABLE klienci (  
  pesel CHAR(11) PRIMARY KEY,  
  adres CHARACTER VARYING(100),  
  wiek NUMERIC(2),  
  wspolpraca_od DATE  
);
```

```
INSERT INTO klienci (pesel) VALUES ('01018110203');  
INSERT INTO klienci (pesel) VALUES ('01018110203');
```

Slajd pokazuje rozwiązanie zadania (2), którego treść przytoczono poniżej.

Utwórz relację KLIENCI o następujących atrybutach i ograniczeniach:

PESEL

Typ łańcuchowy o stałej długości (11 znaków). Klucz podstawowy (ograniczenie atrybutu).

ADRES

Typ łańcuchowy o zmiennej długości (max 100 znaków).

WIEK

Typ liczbowy, (max 2 cyfry).

WSPOLPRACA_OD

Typ reprezentujący datę.

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „klucz podstawowy”.



Wartość unikalna

```

1 CREATE TABLE projekty (
    numer NUMERIC(6) PRIMARY KEY,
    nazwa CHARACTER VARYING (100) UNIQUE,
    budżet NUMERIC(10,2),
    termin_zak DATE
);

```

```

2 CREATE TABLE projekty (
    numer NUMERIC(6),
    nazwa CHARACTER VARYING (100),
    budżet NUMERIC(10,2),
    termin_zak DATE,
    CONSTRAINT un_nazwa UNIQUE (numer,nazwa)
);

```

Ćwiczenie 7 - DDL (19)

Ograniczenie integralnościowe „wartość unikalna” przypomina swoim działaniem ograniczenie „klucz podstawowy”. Żadna kombinacja wartości atrybutów wchodzących w skład ograniczenia nie może wystąpić w relacji więcej niż raz. Dopuszczalne jest jednak, aby wartości tych atrybutów przyjmowały wartości puste. Ograniczenie „wartość unikalna” jako ograniczenie atrybutu definiuje się podając słowo kluczowe **UNIQUE** po opisie atrybutu, którego dotyczy ograniczenie. Podobnie jak z każdym innym ograniczeniem, tutaj również można poprzedzić właściwą definicję ograniczenia słowem kluczowym **CONSTRAINT** i nazwą ograniczenia. Ograniczenie „wartość unikalna” jako ograniczenie relacji definiuje się podając, jako osobny wpis na liście atrybutów, (opcjonalnie) słowo kluczowe **CONSTRAINT** i nazwę ograniczenia, następnie słowo kluczowe **UNIQUE** i w nawiasach listę atrybutów, których dotyczy ograniczenie.

Przykład (1) pokazuje sposób definicji ograniczenia „wartość unikalna” jako ograniczenia atrybutu. W tworzonej na tym przykładzie relacji ograniczenie nadano atrybutowi *nazwa*. Dzięki ograniczeniu, każdy projekt musi mieć inną nazwę, ale możliwe jest wstawienie kilku projektów o nazwie równej **NULL**.

Przykład (2) pokazuje sposób definicji ograniczenia „wartość unikalna” jako ograniczenia relacji. W tworzonej na tym przykładzie relacji ograniczenie nadano parze atrybutów **NUMER** i **NAZWA**. Jak łatwo zauważyć, w tym przykładzie nadano ograniczeniu integralnościowemu nazwę **UN_NAZWA**. Dzięki temu ograniczeniu, żadna kombinacja wartości atrybutu **NUMER** i atrybutu **NAZWA** nie może się powtórzyć. Przykładowo, może wystąpić wiele krotek o wartości atrybutu **NUMER** równej 1, ale muszą one mieć różne **NAZWY**. Może również wystąpić wiele krotek o wartości atrybutu **NAZWA** równej „System informatyczny dla dziekanatu”, jednak wówczas muszą one mieć różne wartości na atrybucie **NUMER**.

Wyjątkiem jest tutaj sytuacja, w której krotka zawiera wartości NULL zarówno w atrybucie NUMER jak i atrybucie NAZWA. Takie krotki nie podlegają ograniczeniu i można wstawić ich dowolną liczbę.

W przeciwieństwie do ograniczenia integralnościowego „klucz podstawowy” ograniczeń „wartość unikalna” można zdefiniować dowolną liczbę.



Zadanie (3)

- Utwórz relację UCZELNIE o następujących atrybutach i ograniczeniach:

Nazwa	Typ
ID_UCZELNI	Liczbowy, (max 4 cyfry). Klucz podstawowy (ograniczenie relacji).
NAZWA	Łańcuchowy o zmiennej długości (max 100 znaków). Wartość unikalna (ograniczenie relacji).
ADRES	Łańcuchowy o zmiennej długości (max 100 znaków)
BUDZET	Liczbowy (max 10 cyfr, 2 po przecinku).
ZALOZONA	Typ reprezentujący datę.

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „wartość unikalna”.



Rozwiązanie (3)

```
CREATE TABLE uczelnie (  
  id_uczelni NUMERIC(4),  
  nazwa CHARACTER VARYING(100),  
  adres CHARACTER VARYING(100),  
  budzet NUMERIC (10,2),  
  zalozona DATE,  
  PRIMARY KEY(id_uczelni),  
  UNIQUE(nazwa)  
);
```

```
INSERT INTO uczelnie (id_uczelni,nazwa) VALUES (10,'PP');  
INSERT INTO uczelnie (id_uczelni,nazwa) VALUES (20,'PP');
```

Ćwiczenie 7 - DDL (22)

Slajd pokazuje rozwiązanie zadania (3), którego treść przytoczono poniżej.

Utwórz relację UCZELNIE o następujących atrybutach i ograniczeniach:

ID_UCZELNI

Typ liczbowy, (max 4 cyfry). Klucz podstawowy (ograniczenie relacji).

NAZWA

Łańcuchowy o zmiennej długości (max 100 znaków). Wartość unikalna (ograniczenie relacji).

ADRES

Łańcuchowy o zmiennej długości (max 100 znaków)

BUDZET

Liczbowy (max 10 cyfr, 2 po przecinku).

ZALOZONA

Typ reprezentujący datę.

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „wartość unikalna”.



Wartość obowiązkowa

```
1 CREATE TABLE projekty (  
    numer NUMERIC(6) PRIMARY KEY,  
    nazwa CHARACTER VARYING (100) UNIQUE NOT NULL,  
    budżet NUMERIC(10,2) CONSTRAINT nn_budżet NOT NULL,  
    termin_zak DATE NOT NULL  
);
```

Ćwiczenie 7 - DDL (23)

Ograniczenie „klucz podstawowy” powodowało, że nie można było podać wartości NULL dla żadnego atrybutu wchodzącego w skład tego ograniczenia. Nie trzeba jednak definiować klucza podstawowego, aby wymusić podawanie wartości różnych od NULL. Istnieje również ograniczenie „wartość obowiązkowa”, z którym nie jest związane wymaganie unikalności wartości atrybutów. Ograniczenie „wartość obowiązkowa” można zdefiniować jedynie jako ograniczenie atrybutu podając słowa kluczowe NOT NULL za właściwą definicją atrybutu (opcjonalnie poprzedzając je nazwą).

Przykład (1) na slajdzie pokazuje kolejną wersję relacji PROJEKTY, w której ograniczenie „wartość obowiązkowa” zdefiniowano dla atrybutów NAZWA, BUDZET i TERMIN_ZAK. Szczególnie interesującym przypadkiem jest tutaj ograniczenie atrybutu NAZWA. Jak łatwo zauważyć zdefiniowano tutaj dwa ograniczenia dla jednego atrybutu. W ogólności możliwe jest zdefiniowanie dowolnej liczby ograniczeń dla jednego atrybutu, dla każdego podając w razie konieczności nazwę. Drugą ciekawą rzeczą jest to, iż zdefiniowano tutaj ograniczenia UNIQUE i NOT NULL, co w efekcie ma takie samo działanie jak ograniczenie PRIMARY KEY.



Zadanie (4)

- Utwórz relację KSIĄZKI o następujących atrybutach i ograniczeniach:

Nazwa	Typ
ID_KSIĄZKI	Liczbowy, (max 10 cyfr). Klucz podstawowy (ograniczenie atrybutu).
TYTUL	Łańcuchowy o zmiennej długości (max 100 znaków). Wartość obowiązkowa
AUTORZY	Łańcuchowy o zmiennej długości (max 100 znaków)
CENA	Liczbowy (max 6 cyfr, 2 po przecinku).
DATA_WYDANIA	Typ reprezentujący datę.

Ćwiczenie 7 - DDL (24)

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „wartość obowiązkowa”.



Rozwiązanie (4)

```
CREATE TABLE ksiazki (  
  id_ksiazki NUMERIC (10) PRIMARY KEY,  
  tytul CHARACTER VARYING (100) NOT NULL,  
  autorzy CHARACTER VARYING (100),  
  cena NUMERIC (6,2),  
  data_wydania DATE  
);
```

```
INSERT INTO ksiazki(id_ksiazki,tytul) VALUES (10,NULL);
```

Ćwiczenie 7 - DDL (25)

Slajd pokazuje rozwiązanie zadania (4), którego treść przytoczono poniżej.

Utwórz relację KSIĄZKI o następujących atrybutach i ograniczeniach:

ID_KSIĄZKI

Typ liczbowy, (max 10 cyfr). Klucz podstawowy (ograniczenie atrybutu).

TYTUL

Typ łańcuchowy o zmiennej długości (max 100 znaków). Wartość obowiązkowa.

AUTORZY

Typ łańcuchowy o zmiennej długości (max 100 znaków).

CENA

Typ liczbowy (max 6 cyfr, 2 po przecinku).

DATA_WYDANIA

Typ reprezentujący datę.

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „wartość obowiązkowa”.



Klucz obcy

```

1 CREATE TABLE projekty (
    nazwa CHARACTER VARYING (100) PRIMARY KEY,
    budżet NUMERIC(10,2),
    termin_zak DATE,
    id_szefa NUMBER (4) REFERENCES pracownicy(id_prac)
);

2 CREATE TABLE projekty (
    nazwa CHARACTER VARYING (100) PRIMARY KEY,
    budżet NUMERIC(10,2),
    termin_zak DATE,
    id_szefa NUMBER (4),
    FOREIGN KEY (id_szefa) REFERENCES pracownicy(id_prac)
);
  
```

Ćwiczenie 7 - DDL (26)

Dotychczas omówione ograniczenia integralnościowe dotyczyły tylko jednej relacji. Istnieje również ograniczenie integralnościowe pozwalające na definiowanie związków pomiędzy relacjami. Ograniczenie to nazywane jest „kluczem obcym”. Przypomnijmy sobie relacje ZESPOLY i PRACOWNICY. Obie relacje posiadają atrybut ID_ZESP. Atrybut ten, w relacji ZESPOLY, pozwala na jednoznaczny identyfikację zespołu. Jak się zapewne już Państwo domyślicie, jest to atrybut, który jest kluczem podstawowym tej relacji. W relacji PRACOWNICY, atrybut ID_ZESP określa jedynie numer zespołu, w którym zatrudniony jest pracownik. Zwróćcie Państwo uwagę na fakt, iż atrybut ten powinien przyjmować jedynie wartości, które występują w atrybucie ID_ZESP relacji ZESPOLY. Ponieważ atrybut ID_ZESP relacji PRACOWNICY powinien przyjmować jedynie wartości klucza podstawowego z innej relacji, nazywa się go kluczem obcym. W ogólności, możliwe jest powiązanie klucza obcego nie tylko z kluczem podstawowym innej relacji, ale także z atrybutami wchodzącymi w skład ograniczenia „wartość unikalna”. Zdefiniowanie ograniczenia integralnościowego „klucz obcy” dla jakiegoś zestawu atrybutów wiąże się wskazaniem klucza podstawowego/wartości unikalnych innej relacji, z którymi dany klucz obcy jest powiązany. Ograniczenie pilnuje, aby nigdy w atrybucie/atributach wchodzących w skład klucza obcego nie pojawiła się kombinacja wartości nie występująca w relacji z kluczem głównym/wartościami unikalnymi. Jedynym wyjątkiem jest sytuacja, kiedy dla jednego lub więcej atrybutów wchodzących w skład klucza obcego poda się wartości NULL. Taka kombinacja wartości nie musi występować w relacji z kluczem głównym/wartościami unikalnymi. W przykładowych relacjach PRACOWNICY i ZESPOLY, nie będzie możliwe wstawienie pracownika przydzielonego do zespołu, który nie istnieje (nie został wstawiony wcześniej do relacji ZESPOLY).

Ograniczenie atrybutu „klucz obcy” definiuje się podając za definicją atrybutu, którego dotyczy ograniczenie, słowo kluczowe REFERENCES, a następnie nazwę relacji z kluczem podstawowym/wartością unikalną i w nawiasie listę atrybutów tej relacji, która wchodzi w skład klucza podstawowego/wartości unikalnej. Ograniczenie relacji „klucz obcy” definiuje się podając słowa kluczowe FOREIGN KEY, następnie, w nawiasie, listę atrybutów wchodzących w skład klucza obcego, słowo kluczowe REFERENCES oraz nazwę relacji z kluczem podstawowym/wartościami unikalnymi i w nawiasie listę atrybutów tej relacji, która wchodzi w skład klucza podstawowego/wartości unikalnych. Ponieważ wartości atrybutów klucza obcego są porównywane z wartościami atrybutów klucza podstawowego/wartości unikalnej w takiej kolejności, jak podano je w definicji klucza obcego, należy zwrócić szczególną uwagę na odpowiednią kolejność.

Przykład (1) przedstawiony na slajdzie definiuje ograniczenie atrybutu „klucz obcy” dla atrybutu ID_SZEFA wskazując na odpowiadający mu klucz podstawowy ID_PRAC w relacji PRACOWNICY. Przykład (2) definiuje taki sam klucz obcy, ale robi to definiując go jako ograniczenie relacji. W przykładzie tym na klucz obcy składa się tylko jeden atrybut, ale w ogólności można by podać większą ich liczbę.

Ograniczenia „klucz obcy” definiowane tak, jak przedstawiono na slajdzie, spowodują, że:

- nie będzie możliwe wstawienie projektu z szefem, który nie został wcześniej wprowadzony do relacji PRACOWNICY,
- nie będzie możliwe usunięcie pracownika, który jest szefem jakiegoś projektu,
- nie będzie możliwa zmiana identyfikatora pracownika, który jest szefem jakiegoś projektu.

Zachowanie to można jednak zmienić. Odpowiednia składnia polecenia CREATE TABLE została opisana na kolejnym slajdzie.



Klucz obcy – cd.

1

```
CREATE TABLE projekty (
  nazwa CHARACTER VARYING (100) PRIMARY KEY,
  budżet NUMERIC(10,2),
  termin_zak DATE,
  id_szefa NUMBER (4) REFERENCES pracownicy(id_prac)
  ON DELETE SET NULL
);
```

2

```
CREATE TABLE projekty (
  nazwa CHARACTER VARYING (100) PRIMARY KEY,
  budżet NUMERIC(10,2),
  id_szefa NUMBER (4),
  FOREIGN KEY (id_szefa) REFERENCES pracownicy(id_prac)
  ON DELETE CASCADE
);
```

Ćwiczenie 7 - DDL (28)

Metodę utrzymywania spójności bazy danych ze względu na ograniczenie integralnościowe „klucz obcy” można zmienić, dodając do definicji ograniczenia słowa kluczowe:

- ON DELETE SET NULL – usunięcie krotki z relacji z kluczem głównym/wartością unikalną spowoduje zapisanie do wskazujących na nią atrybutów tworzących klucz obcy wartości NULL,
- ON DELETE CASCADE. – usunięcie krotki z relacji z kluczem głównym/wartością unikalną spowoduje usunięcie wszystkich krotek, które zawierają klucz obcy wskazujący na usuwaną krotkę,
- ON DELETE SET DEFAULT – usunięcie krotki z relacji z kluczem głównym/wartością unikalną spowoduje zapisanie do wskazujących na nią atrybutów tworzących klucz obcy ich wartości domyślnych,
- ON UPDATE SET NULL – modyfikacja wartości klucza podstawowego/wartości unikalnej w krotce spowoduje zapisanie do wskazujących na nią atrybutów tworzących klucz obcy wartości NULL,
- ON UPDATE CASCADE – modyfikacja wartości klucza podstawowego/wartości unikalnej w krotce spowoduje zapisanie do wskazujących na nią atrybutów tworzących klucz obcy nowej wartości klucza podstawowego/wartości unikalnej,
- ON UPDATE SET DEFAULT - modyfikacja wartości klucza podstawowego/wartości unikalnej w krotce spowoduje zapisanie do wskazujących na nią atrybutów tworzących klucz obcy ich wartości domyślnych.

Dla każdego ograniczenia „klucz obcy” można zdefiniować jedno zachowanie dotyczące usuwania (ON DELETE) i jedno dotyczące modyfikacji (ON UPDATE) (w dowolnej kolejności). W SZBD Oracle zaimplementowano jak na razie jedynie ON DELETE SET NULL i ON DELETE CASCADE.

Rozważmy przykłady przedstawione na slajdzie. W przykładzie (1) zdefiniowano klucz obcy ID_SZEFA wskazujący na klucz podstawowy relacji pracownicy (ID_PRAC) i zmodyfikowano zachowanie tego ograniczenia dodając słowa kluczowe ON DELETE SET NULL. Dzięki temu możliwe będzie usunięcie z relacji PRACOWNICY szefa projektu. Jeżeli usuniemy szefa, to we wszystkich jego projektach atrybutowi ID_SZEFA zostanie przypisane NULL. W przykładzie (2) zdefiniowano klucz obcy ID_SZEFA wskazujący na klucz podstawowy relacji pracownicy (ID_PRAC) i zmodyfikowano zachowanie tego ograniczenia dodając słowa kluczowe ON DELETE CASCADE. Tutaj, usunięcie szefa spowoduje również usunięcie wszystkich jego projektów.



Zadanie (5)

- Utwórz relację POKOJE o następujących atrybutach i ograniczeniach:

Nazwa	Typ
NUMER_POKOJU	Liczbowy (max 3 cyfry). Klucz podstawowy (ograniczenie relacji).
ID_ZESP	Liczbowy (max 2 cyfry). Klucz obcy wskazujący na klucz podstawowy relacji ZESPOLY (ograniczenie atrybutu).
LICZBA_OKIEN	Liczbowy (max 1 cyfra).

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „klucz obcy”.



Rozwiązanie (5)

```
CREATE TABLE pokoje (  
    numer_pokoju NUMERIC (3),  
    id_zesp NUMERIC (2) REFERENCES zespolo(id_zesp),  
    liczba_okien NUMERIC(1),  
    PRIMARY KEY (numer_pokoju)  
);
```

```
INSERT INTO pokoje(numer_pokoju,id_zesp) VALUES (123,90);
```

Slajd pokazuje rozwiązanie zadania (5), którego treść przytoczono poniżej.

Utwórz relację POKOJE o następujących atrybutach i ograniczeniach:

NUMER_POKOJU

Typ liczbowy (max 3 cyfry). Klucz podstawowy (ograniczenie relacji).

ID_ZESP

Typ liczbowy (max 2 cyfry). Klucz obcy wskazujący na klucz podstawowy relacji ZESPOLY (ograniczenie atrybutu).

LICZBA_OKIEN

Typ liczbowy (max 1 cyfra).

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „klucz obcy”.



Ograniczenie domeny atrybutu

1 **CREATE TABLE** projekty (
 numer **NUMERIC(6) PRIMARY KEY**,
 nazwa **CHARACTER VARYING (100) UNIQUE**,
 budżet **NUMERIC(10,2) CHECK (budżet < 1000000)**,
 termin_zak **DATE**
);

2 **CREATE TABLE** projekty (
 numer **NUMERIC(6) PRIMARY KEY**,
 nazwa **CHARACTER VARYING (100) UNIQUE**,
 budżet **NUMERIC(10,2) NOT NULL**,
 termin_rozp **DATE NOT NULL**,
 termin_zak **DATE NOT NULL**,
CHECK (termin_rozp < termin_zak)
);

Ćwiczenie 7 - DDL (32)

Ostatnim ograniczeniem integralnościowym, które można zdefiniować, jest „ograniczenie domeny atrybutu”. Jest nim warunek logiczny zdefiniowany na wartościach atrybutów krotki, który musi być zawsze spełniony. „Ograniczenie domeny atrybutu” jako ograniczenie atrybutu jest definiowane poprzez podanie, za opisem atrybutu, słowa kluczowego **CONSTRAINT** i nazwy ograniczenia (opcjonalnie), a następnie słowa kluczowego **CHECK** i w nawiasie warunku logicznego, który musi być spełniony przez wartość atrybutu. Ważne jest, aby warunek ten dotyczył tylko atrybutu, dla którego zdefiniowane jest ograniczenie. „Ograniczenie domeny atrybutu” jako ograniczenie relacji jest definiowane poprzez podanie słowa kluczowego **CONSTRAINT** i nazwy ograniczenia (opcjonalnie), a następnie słowa kluczowego **CHECK** i w nawiasie warunku logicznego, który muszą spełniać wartości atrybutów relacji. W przypadku ograniczenia relacji możliwe jest podawanie warunków dotyczących więcej niż jednego atrybutu relacji.

Rozważmy przykłady na slajdzie. W przykładzie (1) zdefiniowano ograniczenie atrybutu ograniczające domenę atrybutu **BUDZET** do wartości liczbowych mniejszych niż 1000000. W przykładzie (2) zdefiniowano ograniczenie relacji mówiące o tym, by termin rozpoczęcia projektu był mniejszy od terminu jego zakończenia.



Zadanie (6)

- Utwórz relację PLYTY_CD o następujących atrybutach i ograniczeniach:

Nazwa	Typ
KOMPOZYTOR	Łańcuchowy (max 100 znaków). Wartość obowiązkowa.
TYTUL_ALBUMU	Łańcuchowy (max 100 znaków). Wartość obowiązkowa. Kombinacja wartości kompozytora i tytułu albumu musi być unikalna (nadaj ograniczeniu nazwę UN_KO_TY).
DATA_NAGRANIA	Typ reprezentujący datę.
DATA_WYDANIA	Typ reprezentujący datę. Data nagrania musi być wcześniejsza niż data wydania (ograniczenie relacji).
CZAS_TRWANIA	Typ reprezentujący okres czasu (INTERVAL). Czas musi być krótszy niż 82 minuty (ograniczenie atrybutu).

Ćwiczenie 7 - DDL (33)

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „ograniczenie domeny atrybutu”.

(!) W zadaniu nakazano użyć typu INTERVAL. Normalnie należałoby użyć tutaj typu INTERVAL MINUTE(2) TO SECOND. Jeżeli do wykonywania ćwiczeń korzystacie państwo z SZBD Oracle, należy użyć tutaj bardziej ogólnego typu INTERVAL DAY TO SECOND, gdyż typ INTERVAL MINUTE TO SECOND nie jest wspierany.



Rozwiązanie (6)

```
CREATE TABLE plyty_cd (  
  kompozytor CHARACTER VARYING (100) NOT NULL,  
  tytuł_albumu CHARACTER VARYING (100) NOT NULL,  
  CONSTRAINT un_komp_album UNIQUE (kompozytor,tytuł_albumu),  
  data_nagrania DATE,  
  data_wydania DATE,  
  CHECK (data_nagrania<data_wydania),  
  czas_trwania INTERVAL DAY TO SECOND,  
  check(czas_trwania < INTERVAL '82:0' MINUTE(2) TO SECOND)  
);
```

Ćwiczenie 7 - DDL (34)

Slajd pokazuje rozwiązanie zadania (6), którego treść przytoczono poniżej. Rozwiązanie jest kontynuowane na kolejnym slajdzie.

Utwórz relację PLYTY_CD o następujących atrybutach i ograniczeniach:

KOMPOZYTOR

Typ łańcuchowy (max 100 znaków). Wartość obowiązkowa.

TYTUL_ALBUMU

Typ łańcuchowy (max 100 znaków). Wartość obowiązkowa. Kombinacja wartości kompozytora i tytułu albumu musi być unikalna (nadaj ograniczeniu nazwę UN_KO_TY).

DATA_NAGRANIA

Typ reprezentujący datę.

DATA_WYDANIA

Typ reprezentujący datę. Data nagrania musi być wcześniejsza niż data wydania (ograniczenie relacji).

CZAS_TRWANIA

Typ reprezentujący okres czasu (INTERVAL). Czas musi być krótszy niż 82 minuty (ograniczenie atrybutu).

Po utworzeniu relacji spróbuj wstawić krotki naruszające ograniczenie integralnościowe „ograniczenie domeny atrybutu”.



Rozwiązanie (6) – cd.

```
INSERT INTO plyty_cd(kompozytor, tytul_albumu, data_nagrania,  
data_wydania, czas_trwania)  
VALUES ('Queen', 'Flash Gordon', DATE '1980-01-01',  
DATE '1994-01-01',INTERVAL '83:0' MINUTE(2) TO SECOND);
```



Tworzenie relacji przez podzapytanie

```

1 CREATE TABLE nazwa_relacji
      (nazwa_atrybutu [DEFAULT wartość_domyślna]
        [ [CONSTRAINT nazwa_ogr] ograniczenie_atr]...,
       nazwa_atrybutu [DEFAULT wartość_domyślna]
        [ [CONSTRAINT nazwa_ogr] ograniczenie_atr]...,
       ....
       [ [CONSTRAINT nazwa_ogr] ograniczenie_rel, ...] )
  AS SELECT zapytanie;

2 CREATE TABLE roczne_place (nazwisko PRIMARY KEY, etat, placa)
  AS SELECT nazwisko, etat, 12 * (placa_pod + NVL(placa_dod,0))
  FROM pracownicy;

```

Ćwiczenie 7 - DDL (36)

Polecenie CREATE TABLE tworzące relację można rozbudować o możliwość podania zapytania, którego wyniki posłużą do wypełnienia nowo utworzonej relacji. W tej wersji polecenia CREATE TABLE, na liście atrybutów nie podajemy typów atrybutów, a jedynie wartości domyślne i ograniczenia integralnościowe. Zapytanie wypełniające relację jest podawane po słowie kluczowym AS za listą atrybutów. Jeżeli wynik zapytania nie spełnia ograniczeń integralnościowych zdefiniowanych na liście atrybutów, próba utworzenia relacji kończy się błędem. Ogólną składnię rozbudowanego polecenia przedstawiono na przykładzie (1). Ważne jest tutaj, aby liczba atrybutów w relacji wynikowej zapytania była taka sama jak liczba wyrażeń zdefiniowanych w klauzuli SELECT. Rozważmy przykład (2). Przykład ten przedstawia polecenie tworzące relację ROCZNE_PLACE z atrybutami NAZWISKO (z ograniczeniem „klucz podstawowy”), ETAT i PLACA, i zapisujące do tej relacji wynik zapytania, które zwraca nazwisko, etat i roczną płacę wszystkich pracowników.

Możliwe jest również pominięcie listy atrybutów i podanie tylko zapytania. Wówczas tworzona jest relacja z atrybutami o nazwach i typach takich jakie są zawarte w relacji wynikowej zapytania. Przykładowo, wersja polecenia z przykładu (2), w której nie definiuje się listy atrybutów wygląda następująco:

```

CREATE TABLE roczne_place
AS SELECT nazwisko, etat, 12 * (placa_pod + NVL(placa_dod,0)) as placa
FROM pracownicy;

```

Oczywiście, w tej wersji polecenia CREATE TABLE nie jest możliwe definiowanie ograniczeń integralnościowych. Możliwe jest jednak późniejsze ich dodanie za pomocą polecenia ALTER TABLE opisanego na kolejnych slajdach.



Zadanie (7)

- Utwórz relację zawierającą nazwisko szefa i nazwisko podwładnego, i wypełnij ją danymi (całość wykonaj za pomocą jednego polecenia).



Rozwiązanie (7)

```
CREATE TABLE prac_szef (szef, podwladny) AS  
SELECT s.nazwisko, p.nazwisko  
FROM pracownicy s JOIN pracownicy p ON (s.id_prac=p.id_szefa);
```

Slajd pokazuje rozwiązanie zadania (7), którego treść przytoczono poniżej.

Utwórz relację zawierającą nazwisko szefa i nazwisko podwładnego, i wypełnij ją danymi (całość wykonaj za pomocą jednego polecenia).



Modyfikowanie schematu relacji

1 **ALTER TABLE** nazwa_relacji
ADD [nazwa typ(rozmiar) [DEFAULT wartość] ograniczenia |
CONSTRAINT nazwa typ ograniczenie];

2 **ALTER TABLE** nazwa_relacji
MODIFY (nazwa typ(rozmiar) [DEFAULT wartość] ograniczenia);

3 **ALTER TABLE** nazwa_relacji
DROP [**COLUMN** (nazwa) | **CONSTRAINT** (nazwa)];

Ćwiczenie 7 - DDL (39)

Polecenie ALTER TABLE pozwala na modyfikowanie relacji. Dzięki niemu możliwe jest dodawanie i usuwanie atrybutów i ograniczeń integralnościowych oraz modyfikowanie definicji atrybutu. Przykład (1) pokazuje składnię polecenia ALTER TABLE służącą do dodawania nowych atrybutów lub ograniczeń integralnościowych. Polecenie to rozpoczyna się od słów kluczowych ALTER TABLE, po których podaje się nazwę relacji, słowo kluczowe ADD i: albo definicję atrybutu (łącznie z ograniczeniami atrybutu), albo definicję ograniczenia relacji. Przykładowo, polecenie:

```
ALTER TABLE pracownicy ADD numer_telefonu CHARACTER VARYING (20) CHECK
(length(numer_telefonu)>6);
```

spowoduje dodanie do relacji PRACOWNICY atrybutu typu łańcuchowego NUMER_TELEFONU, którego długość nie powinna być mniejsza niż 6 znaków. Z kolei polecenie:

```
ALTER TABLE pracownicy ADD CONSTRAINT un_imie_nazwisko UNIQUE
(imie,nazwisko);
```

spowoduje dodanie do relacji PRACOWNICY ograniczenia relacji o nazwie UN_IMIE_NAZWISKO, którego celem jest zapewnienie unikalności każdej kombinacji imienia i nazwiska.

Przykład (2) pokazuje składnię polecenia ALTER TABLE służącą do modyfikacji definicji atrybutu. Polecenie to rozpoczyna się od słów kluczowych ALTER TABLE, po których podaje się: nazwę relacji, słowo kluczowe MODIFY i nową definicję atrybutu. Ważne jest tutaj, aby atrybut, którego nazwę podaje się w definicji, znajdował się już w relacji.

Przykładowo, polecenie:

```
ALTER TABLE pracownicy MODIFY imie CHARACTER VARYING (100) NOT NULL
```

zmodyfikuje typ atrybutu IMIE relacji PRACOWNICY na CHARACTER VARYING (100) i nada mu ograniczenie atrybutu „wartość obowiązkowa” (NOT NULL).

Przykład (3) pokazuje składnię polecenia ALTER TABLE służącą do usuwania atrybutu lub ograniczenia integralnościowego. Polecenie to rozpoczyna się od słów kluczowych ALTER TABLE, po których podaje się nazwę relacji, słowo kluczowe DROP i: albo słowo kluczowe COLUMN i nazwa atrybutu, albo słowo kluczowe CONSTRAINT i nazwa ograniczenia.

Przykładowo, polecenie:

```
ALTER TABLE pracownicy DROP CONSTRAINT fk_etat;
```

spowoduje usunięcie ograniczenia integralnościowego o nazwie FK_ETAT. W relacji PRACOWNICY jest to nazwa ograniczenia „klucz obcy” wskazującego na atrybut o nazwie NAZWA w relacji ETATY. Polecenie:

```
ALTER TABLE pracownicy DROP COLUMN imie;
```

spowoduje usunięcie atrybutu IMIE z relacji PRACOWNICY.



Zadanie (8)

- Zmień ograniczenie „wartość unikalna” założone na atrybutach kompozytor i tytuł_albumu relacji PLYTY_CD, na ograniczenie „klucz podstawowy”. W tym celu najpierw usuń ograniczenie „wartość unikalna” i załóż nowe ograniczenie: „klucz podstawowy” na tych samych atrybutach.



Rozwiązanie (8)

```
ALTER TABLE plyty_cd DROP CONSTRAINT un_ko_ty;
```

```
ALTER TABLE plyty_cd ADD CONSTRAINT pk_ko_ty  
PRIMARY KEY (kompozytor,tytul_albumu);
```

Ćwiczenie 7 - DDL (42)

Slajd pokazuje rozwiązanie zadania (8), którego treść przytoczono poniżej.

Zmień ograniczenie „wartość unikalna” założone na atrybutach KOMPOZYTOR i TYTUL_ALBUMU relacji PLYTY_CD, na ograniczenie „klucz podstawowy”. W tym celu najpierw usuń ograniczenie „wartość unikalna” i załóż nowe ograniczenie: „klucz podstawowy” na tych samych atrybutach.



1

```
ALTER TABLE relacja  
ENABLE [CONSTRAINT nazwa | rodzaj]
```

2

```
ALTER TABLE pracownicy  
ENABLE CONSTRAINT fk_etat;
```

3

```
ALTER TABLE relacja  
DISABLE [CONSTRAINT nazwa | rodzaj]
```

4

```
ALTER TABLE pracownicy  
DISABLE PRIMARY KEY;
```

Ograniczenia integralnościowe można włączać i wyłączać za pomocą polecenia ALTER TABLE. Wyłączone ograniczenie nie jest w ogóle sprawdzane i mogą być do relacji wstawiane krotki niezgodne z tym ograniczeniem. Próba włączenia ograniczenia może zakończyć się błędem, jeżeli wstawione krotki nie spełniają tego ograniczenia. Przykład (1) pokazuje ogólną składnię polecenia ALTER TABLE pozwalającą na włączenie wcześniej wyłączanego ograniczenia. Polecenie rozpoczyna się od słów kluczowych ALTER TABLE, po których podaje się nazwę relacji, potem słowo kluczowe ENABLE i: albo słowo kluczowe CONSTRAINT i nazwę ograniczenia, albo rodzaj ograniczenia (np. PRIMARY KEY, UNIQUE). Przykład (2) pokazuje polecenie włączające ograniczenie o nazwie FK_ETAT. Przykład (3) demonstruje składnię polecenia ALTER TABLE pozwalającą na wyłączenie ograniczenia. Jest ona identyczna ze składnią pozwalającą na włączanie ograniczeń z tą jedną różnicą, że słowo kluczowe ENABLE zastąpiono słowem kluczowym DISABLE. Przykład (4) pokazuje polecenie wyłączające ograniczenie „klucz podstawowy” relacji PRACOWNICY.

Jeżeli podczas tworzenia relacji nie zdefiniowano nazwy ograniczenia integralnościowego, to nazwa ta jest generowana automatycznie przez SZBD. Jeżeli nie znamy nazwy ograniczenia, to w każdym SZBD istnieje możliwość sprawdzenia, jakie ograniczenia integralnościowe są zdefiniowane dla danej relacji. Sposób odczytania listy nazw ograniczeń zależy od SZBD. W SZBD Oracle można ją odczytać z tzw. słownika bazy danych. Jest to zestaw relacji, które przechowują informacje o różnych aspektach pracy SZBD. Przykładowo, aby odczytać listę nazw ograniczeń, ich typów i, w przypadku ograniczenia integralnościowego „ograniczenie domeny atrybutu” – warunku, można wykorzystać zapytanie:

```
SELECT constraint_name, constraint_type, search_condition  
FROM user_constraints  
WHERE table_name='NAZWA_RELACJI';
```



Zadanie (9)

- Wyłącz ograniczenie „klucz podstawowy” założone na atrybutach kompozytor i tytuł_albumu relacji PLYTY_CD. Wstaw krotki naruszające wyłączone ograniczenie. Spróbuj włączyć ograniczenie. Co się stało?



Rozwiązanie (9)

```
ALTER TABLE plyty_cd DISABLE CONSTRAINT pk_ko_ty;
```

```
INSERT INTO plyty_cd(kompozytor, tytul_albumu,  
data_nagrania, data_wydania, czas_trwania)  
VALUES ('Queen', 'Flash Gordon', DATE '1980-01-01',  
DATE '1994-01-01',INTERVAL '80:0' MINUTE(2) TO SECOND);
```

```
INSERT INTO plyty_cd(kompozytor, tytul_albumu,  
data_nagrania, data_wydania, czas_trwania)  
VALUES ('Queen', 'Flash Gordon', DATE '1980-01-01',  
DATE '1994-01-01',INTERVAL '80:0' MINUTE(2) TO SECOND);
```

```
ALTER TABLE plyty_cd ENABLE CONSTRAINT pk_ko_ty;
```

Ćwiczenie 7 - DDL (46)

Slajd pokazuje rozwiązanie zadania (9), którego treść przytoczono poniżej.

Wyłącz ograniczenie „klucz podstawowy” założone na atrybutach kompozytor i tytul_albumu relacji PLYTY_CD. Wstaw krotki naruszające wyłączone ograniczenie. Spróbuj włączyć ograniczenie. Co się stało?



Zmiana nazwy i usuwanie relacji

- 1 `RENAME stara_nazwa TO nowa_nazwa;`
- 2 `RENAME pracownicy TO zatrudnieni;`
- 3 `DROP TABLE nazwa_relacji [CASCADE CONSTRAINTS];`
- 4 `DROP TABLE etaty CASCADE CONSTRAINTS;`

Ćwiczenie 7 - DDL (47)

Ostatnimi poleceniami, jakie przedstawimy państwu na tych ćwiczeniach, są polecenia `RENAME` i `DROP TABLE` pozwalające odpowiednio na: zmianę nazwy relacji i usunięcie relacji. Przykład (1) pokazuje składnię polecenia `RENAME`. Polecenie to rozpoczyna się od słowa kluczowego `RENAME`, następnie podaje się nazwę modyfikowanej relacji, słowo kluczowe `TO` i nową nazwę relacji. Przykład (2) demonstruje użycie tego polecenia do zmiany nazwy relacji `PRACOWNICY` na nazwę `ZATRUDNIENI`. Przykład (3) pokazuje składnię polecenia `DROP TABLE`. Polecenie rozpoczyna się od słów kluczowych `DROP TABLE`, po których podaje się nazwę usuwanej relacji i opcjonalnie słowa kluczowe `CASCADE CONSTRAINTS`. `CASCADE CONSTRAINTS` oznacza, że usuwając relację należy również usunąć również wszystkie ograniczenia „klucz obcy”, które wskazują na atrybuty usuwanej relacji. W sytuacji, kiedy takie ograniczenia istnieją, ale w poleceniu `DROP TABLE` nie użyje się słów `CASCADE CONSTRAINTS`, to wykonanie polecenie kończy się błędem. Przykład (4) pokazuje użycie tego polecenia do usunięcia relacji `ETATY`.



Zadanie (10)

- Zmień nazwę relacji ZWIERZETA na GATUNKI, a potem ją usuń.



Rozwiązanie (10)

```
RENAME zwierzeta TO gatunki;
```

```
DROP TABLE gatunki;
```

Slajd pokazuje rozwiązanie zadania (10), którego treść przytoczono poniżej.

Zmień nazwę relacji ZWIERZETA na GATUNKI, a potem ją usuń.



11. Utwórz relację PROJEKTY o następujących atrybutach i ograniczeniach:

Nazwa	Typ
ID_PROJEKTU	Liczbowy (max 4 cyfry). Klucz podstawowy.
OPIS_PROJEKTU	Łańcuchowy (max 20 znaków). Wartość obowiązkowa i unikalna.
DATA_ROZPOCZECIA	Typ reprezentujący datę. Domyślnie data systemowa
DATA_ZAKONCZENIA	Typ reprezentujący datę. Późniejsza niż data rozpoczęcia.
FUNDUSZ	Liczbowy (max 7 cyfr, w tym 2 po przecinku).



Zadania

12. Utwórz relację PRZYDZIAŁY o następujących atrybutach i ograniczeniach

Nazwa	Typ
ID_PROJEKTU	Liczbowy (max 4 cyfry). Klucz obcy (PROJEKTY).
ID_PRAC	Liczbowy (max 4 cyfry). Klucz obcy (PRACOWNICY).
OD	Typ reprezentujący datę. Domyślnie data systemowa
DO	Typ reprezentujący datę. Data późniejsza niż OD.
STAWKA	Liczbowy (max 7 cyfr, w tym 2 po przecinku).
ROLA	Łańcuchowy (max 20 znaków). Jedynie wartości: KIERUJACY, ANALITYK i PROGRAMISTA.
Klucz podstawowy tworzą atrybuty ID_PROJEKTU i ID_PRAC.	



Zadania

13. Dodaj do relacji PRZYDZIAŁY atrybut GODZINY typu liczbowego.
14. Wyłącz tymczasowo sprawdzanie unikalności opisów projektów.
15. Zwiększ maksymalny rozmiar atrybutu OPIS_PROJEKTU do 30 znaków.
16. Wstaw pracownika o nazwisku Kowalski do relacji PRACOWNICY. Spróbuj założyć ograniczenie „wartość unikalna” na atrybucie nazwisko. Co się stało?
17. Usuń atrybut imie z relacji PRACOWNICY.



Zadania

18. Utwórz relację PRACOWNICY_ZESPOLY zawierającą następujące atrybuty: NAZWISKO, ETAT, ROCZNA_PLACA, ZESPOL i ADRES_PRACY, i wypełnij ją korzystając z mechanizmu tworzenia relacji w oparciu o zapytanie.



11

```
CREATE TABLE projekty (  
  id_projektu NUMERIC (4) PRIMARY KEY,  
  opis_projektu CHARACTER VARYING (20)  
  CONSTRAINT un_opis UNIQUE NOT NULL,  
  data_roz poczenia DATE DEFAULT SYSDATE,  
  data_zakonczenia DATE,  
  CHECK (data_roz poczenia < data_zakonczenia),  
  fundusz NUMERIC(7,2)  
);
```

Slajd pokazuje rozwiązanie zadania (11), którego treść przytoczono poniżej.

Utwórz relację PROJEKTY o następujących atrybutach i ograniczeniach:

ID_PROJEKTU

Typ liczbowy (max 4 cyfry). Klucz podstawowy.

OPIS_PROJEKTU

Łańcuchowy (max 20 znaków). Wartość obowiązkowa i unikalna.

DATA_ROZPOCZECIA

Typ reprezentujący datę. Domyślnie data systemowa

DATA_ZAKONCZENIA

Typ reprezentujący datę. Późniejsza niż data rozpoczęcia.

FUNDUSZ

Typ liczbowy (max 7 cyfr, w tym 2 po przecinku).



```

12 CREATE TABLE przydzialy (
  id_projektu NUMERIC(4) REFERENCES projekty (id_projektu),
  id_prac NUMERIC(4) REFERENCES pracownicy (id_prac),
  od DATE DEFAULT SYSDATE,
  do DATE,
  CHECK (od<do),
  stawka NUMERIC(7,2),
  rola CHARACTER VARYING (20)
  CHECK (rola IN ('KIERUJACY', 'ANALITYK', 'PROGRAMISTA')),
  PRIMARY KEY (id_projektu,id_prac)
);

```

Slajd pokazuje rozwiązanie zadania (12), którego treść przytoczono poniżej.

Utwórz relację PRZYDZIALY o następujących atrybutach i ograniczeniach

ID_PROJEKTU

Typ liczbowy (max 4 cyfry). Klucz obcy (PROJEKTY).

ID_PRAC

Typ liczbowy (max 4 cyfry). Klucz obcy (PRACOWNICY).

OD

Typ reprezentujący datę. Domyślnie data systemowa

DO

Typ reprezentujący datę. Data późniejsza niż OD.

STAWKA

Typ liczbowy (max 7 cyfr, w tym 2 po przecinku).

ROLA

Typ łańcuchowy (max 20 znaków). Jedynie wartości: KIERUJACY, ANALITYK i PROGRAMISTA.

Klucz podstawowy tworzą atrybuty ID_PROJEKTU i ID_PRAC.



13 ALTER TABLE przydzialy ADD godziny NUMERIC(3);

14 ALTER TABLE projekty DISABLE CONSTRAINT un_opis;

15 ALTER TABLE projekty MODIFY opis_projektu
CHARACTER VARYING (30);

16 INSERT INTO pracownicy(id_prac,nazwisko)
VALUES(300,'Kowalski');

ALTER TABLE pracownicy ADD UNIQUE(nazwisko);

Slajd pokazuje rozwiązania zadań (13), (14), (15) i (16), których treść przytoczono poniżej.

- (13) Dodaj do relacji PRZYDZIAŁY atrybut GODZINY typu liczbowego.
- (14) Wyłącz tymczasowo sprawdzanie unikalności opisów projektów.
- (15) Zwiększ maksymalny rozmiar atrybutu OPIS_PROJEKTU do 30 znaków.
- (16) Wstaw pracownika o nazwisku Kowalski do relacji PRACOWNICY. Spróbuj założyć ograniczenie „wartość unikalna” na atrybucie nazwisko. Co się stało?



17 **ALTER TABLE** pracownicy **DROP COLUMN** imie;

18 **CREATE TABLE** pracownicy_zespoly (nazwisko, etat,
roczna_placa, zespoly, adres_pracy) **AS**
SELECT nazwisko, etat,placa_pod*12, nazwa, adres
FROM pracownicy **NATURAL JOIN** zespoly;

Slajd pokazuje rozwiązania zadań (17) i (18), których treść przytoczono poniżej.

- (17) Usuń atrybut imie z relacji PRACOWNICY.
- (18) Utwórz relację PRACOWNICY_ZESPOLY zawierającą następujące atrybuty: NAZWISKO, ETAT, ROCZNA_PLACA, ZESPOL i ADRES_PRACY, i wypełnij ją korzystając z mechanizmu tworzenia relacji w oparciu o zapytanie.



Podsumowanie

```
CREATE TABLE nazwa_relacji  
    (nazwa_atrybutu typ (rozmiar) [DEFAULT wartość_domyślna]  
      [ [CONSTRAINT nazwa_ogr] ograniczenie_atr],  
    nazwa_atrybutu typ (rozmiar) [DEFAULT wartość_domyślna]  
      [ [CONSTRAINT nazwa_ogr] ograniczenie_atr],  
    ....  
    [ [CONSTRAINT nazwa_ogr] ograniczenie_rel, ...] );
```

①

②

ALTER TABLE

③

RENAME

④

DROP TABLE

⑤

PRIMARY KEY

⑥

UNIQUE

⑦

NOT NULL

⑧

CHECK

⑨

FOREIGN KEY ... REFERENCES

Ćwiczenie 7 - DDL (58)

Na ćwiczeniu zapoznali się Państwo z poleceniami DDL pozwalającymi na tworzenie (1), modyfikację (2), zmianę nazwy (3) i usuwanie (4) relacji. Poznaliście podstawowe typy danych, jakie można wykorzystać do tworzenia relacji i poznaliście różne typy ograniczeń integralnościowych, w tym klucz podstawowy (5), wartość unikalna (6), wartość obowiązkowa (7), ograniczenie domeny atrybutu (8) i klucz obcy (9). Omówioną tematykę przećwiczyliście państwo na zadaniach do samodzielnego wykonania.