



**Wydział Elektroniki
i Technik Informatycznych**

POLITECHNIKA WARSZAWSKA

Bazy Danych 1

edycja 21L

Laboratorium 2

Wprowadzenie i przebieg laboratorium

 Dokończenie generowania diagramów ER oraz relacyjnego z poprzednich ćwiczeń

 DDL: polecenia CREATE i ALTER

 Ćwiczenia z tworzenia tabel oraz modyfikowania ich struktury

 DDL: polecenia RENAME, DROP, FLASHBACK

 Przedstawienie pracy domowej

Zastosowanie języka DDL do tworzenia
schematu bazy danych

SQL

Służy do wydawania poleceń do DBMS. Język deklaratywny.

- **DDL (Data Definition Language)**
 - CREATE, ALTER, DROP, RENAME, TRUNCATE
- **DML (Data Manipulation Language)**
 - INSERT, UPDATE, DELETE
- **DCL(Data Control Language)**
 - GRANT, REVOKE ...
- **TCL (Transaction Control Language)**
 - COMMIT, ROLLBACK, CREATE SAVEPOINT
- **DQL (Data Query Language)**
 - SELECT

W teorii: standard ANSI W praktyce: istnieją różne dialekty SQL.

SQL - szczegóły techniczne

- Wielkość liter w poleceniach bez znaczenia ⚠
 - CREATE TABLE = create table = CREate TaBle
 - przy porównywanie łańcuchów znakowych oczywiście wielkość liter ma znaczenie
- Elementy polecenia można rozdzielać spacją lub nową linią.
- Każde polecenie zakończone średnikiem
- Komentarz: -- treść komentarza lub /* treść komentarza */

✓ [SQL Format - Jak pisać dobry SQL](#)

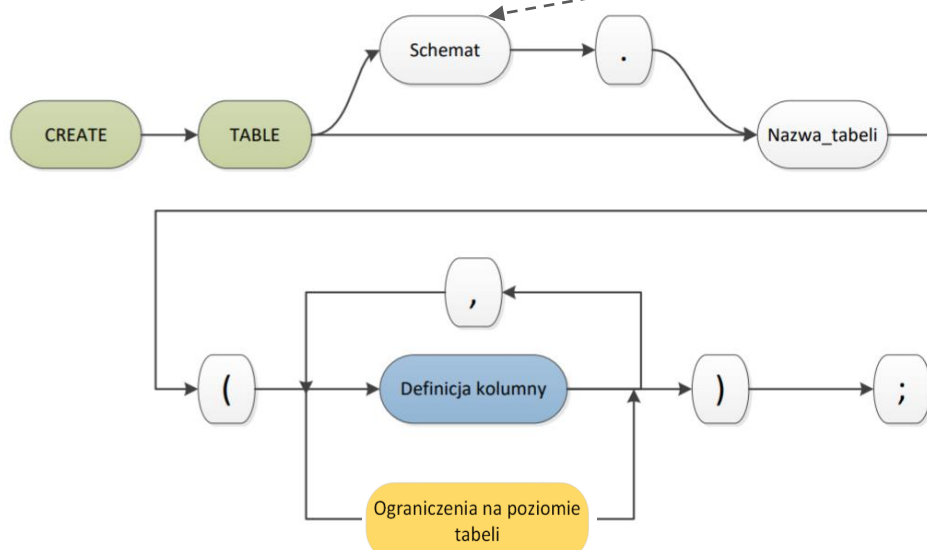
✓ [SQL Style Guide](#)

Data Definition Language (DDL)

- Jest to *zbiór poleceń* pozwalających na tworzenie struktur schematu bazy danych. W szczególności obejmuje takie polecenia jak:
 - **CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, CREATE VIEW** itp.
 - Często wykorzystywany razem z j. *Data Manipulation Language (DML)*.
- DDL a transakcje:
 - Polecenie DDL nie zatwierdzamy poleceniem COMMIT - każde z takich poleceń wykonywane jest zawsze w niezależnej transakcji.
 - wywołanie polecenia DDL wykonuje niejawni COMMIT przed i po wykonaniu polecenia

Polecenie CREATE TABLE

Ogólna sekwencja polecenia jest następująca:



Schemat tożsamy z nazwą użytkownika (opcjonalnie)

Polecenie CREATE TABLE (definicja kolumny)

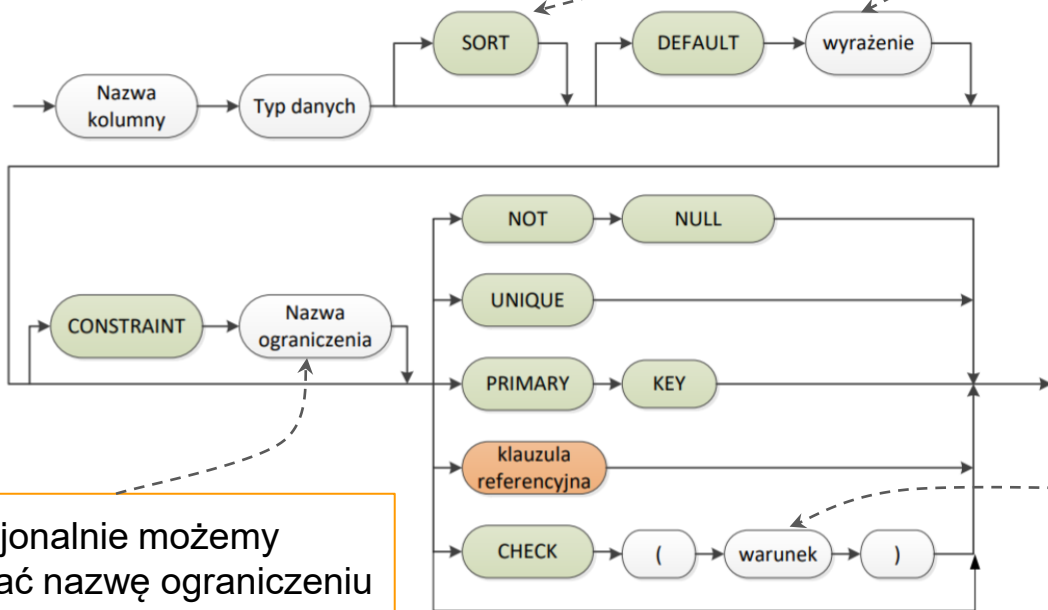
Definicja kolumny jest następująca:

Wskazujemy, czy dane dodawane do kolumny mają być automatycznie sortowane

Wł. domyślna

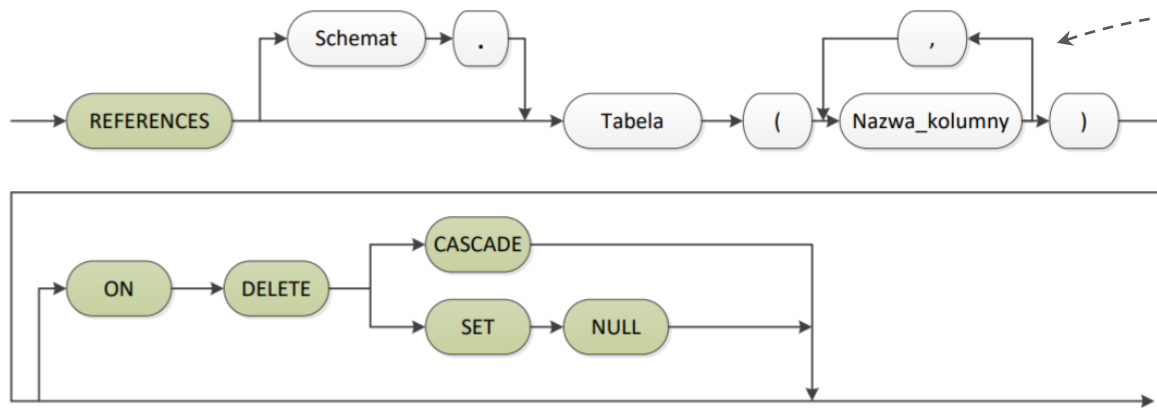
CHECK zawiera warunki logiczne połączone spójnikami logicznymi AND lub OR (w tym miejscu mogą odwoływać się tylko do definiowanej kolumny).

Opcjonalnie możemy nadać nazwę ograniczeniu



Polecenie CREATE TABLE (references - klucz obcy)

Klauzula referencyjna definiująca* klucz obcy może wyglądać następująco:



Klucz obcy może się składać z wielu kolumn (o ile tworzą one klucz główny).

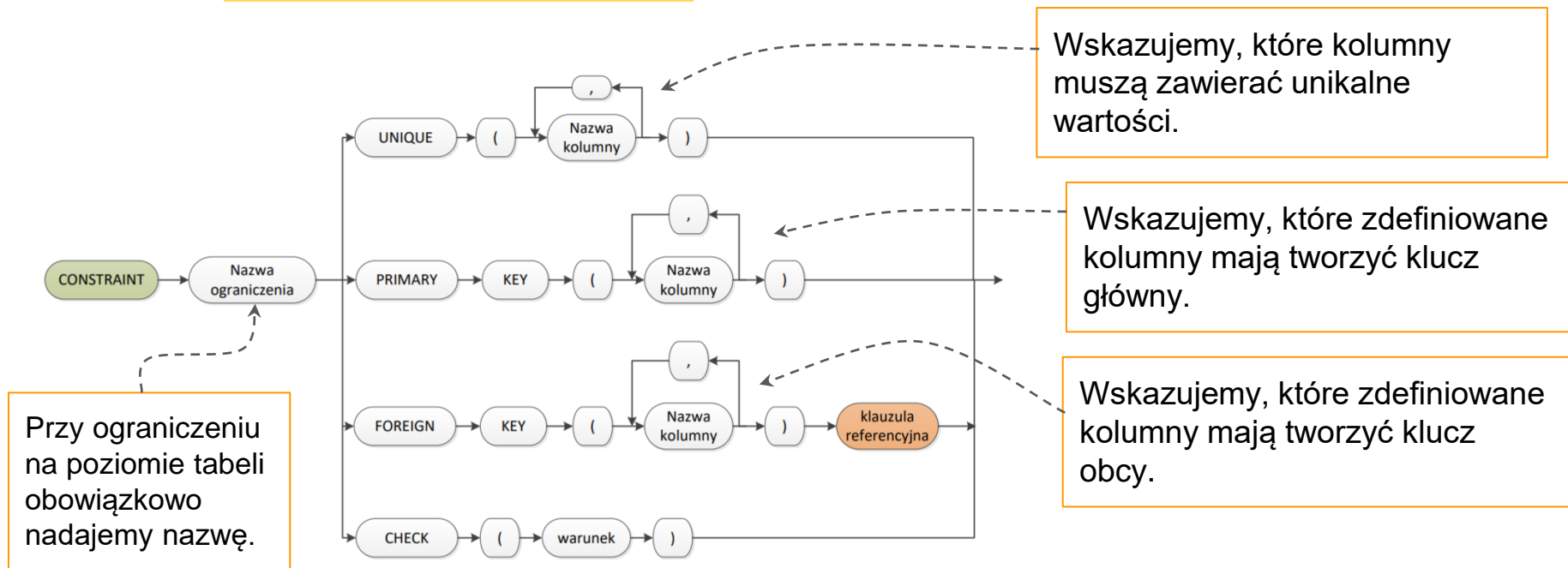
ON DELETE CASCADE - usuwany jest cały wiersz zawierający klucz obcy.

SET NULL - wt. klucza obcego jest ustawiana na NULL

*) poprzez wskazanie klucza głównego w innej tabeli

Polecenie CREATE TABLE (zewnątrzne ograniczenia)

Ewentualne **ograniczenia na poziomie tabeli** mogą być zdefiniowane następująco:



Tworzenie tabeli **Pracownicy** (+ definiowanie klucza głównego)

```
CREATE TABLE pracownicy
(
    id_pracownika      NUMBER (4) CONSTRAINT pracownicy_pk PRIMARY KEY
    , imie              VARCHAR2 (40 CHAR) NOT NULL
    , nazwisko          VARCHAR2 (40 CHAR) NOT NULL
    , zarobki           NUMBER (7, 2)
    , data_zatrudnienia DATE
    , id_zakladu        NUMBER (4) NOT NULL
    , id_kierownika     NUMBER (4)
    , kod_stanowiska    NUMBER (4) NOT NULL
);
```

-- Opis założonej tabeli:

```
DESCRIBE pracownicy;
```

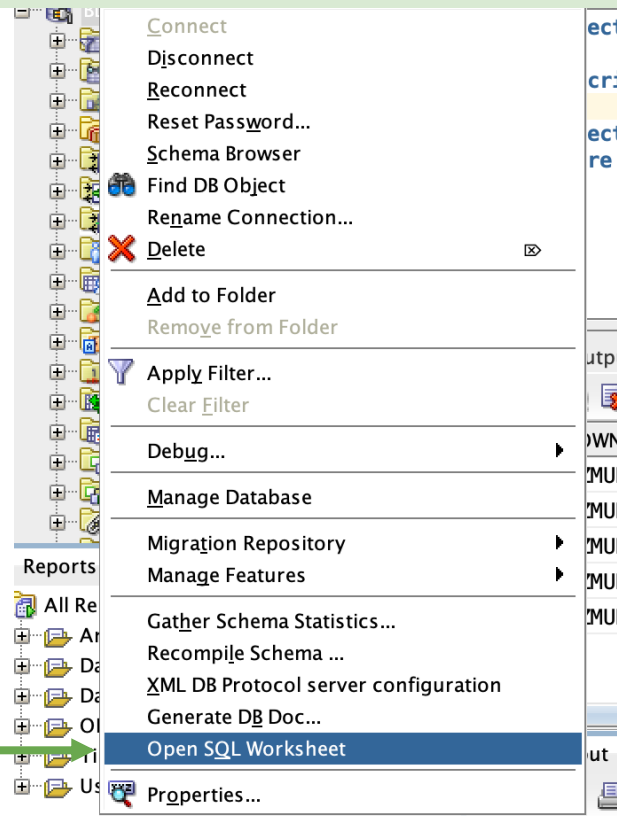
-- Usunięcie tabeli:

```
DROP TABLE pracownicy;
```

SQL Developer

Polecenia SQL piszemy w SQL Worksheet.

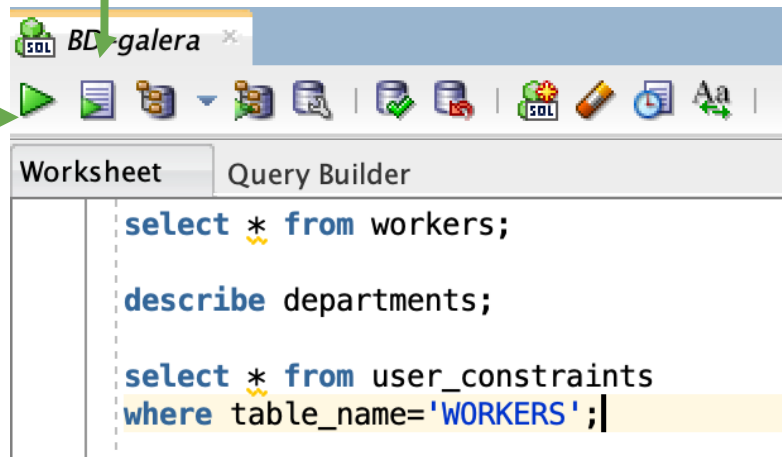
Można mieć otwarty więcej niż 1 worksheet



SQL Developer

Można mieć w worksheet wiele poleceń. Możemy wykonać:

- wiele polecenia z worksheet
 - wszystkie, lub zaznaczone
- pojedyncze polecenie
 - aktywne, to na którym mamy kursor



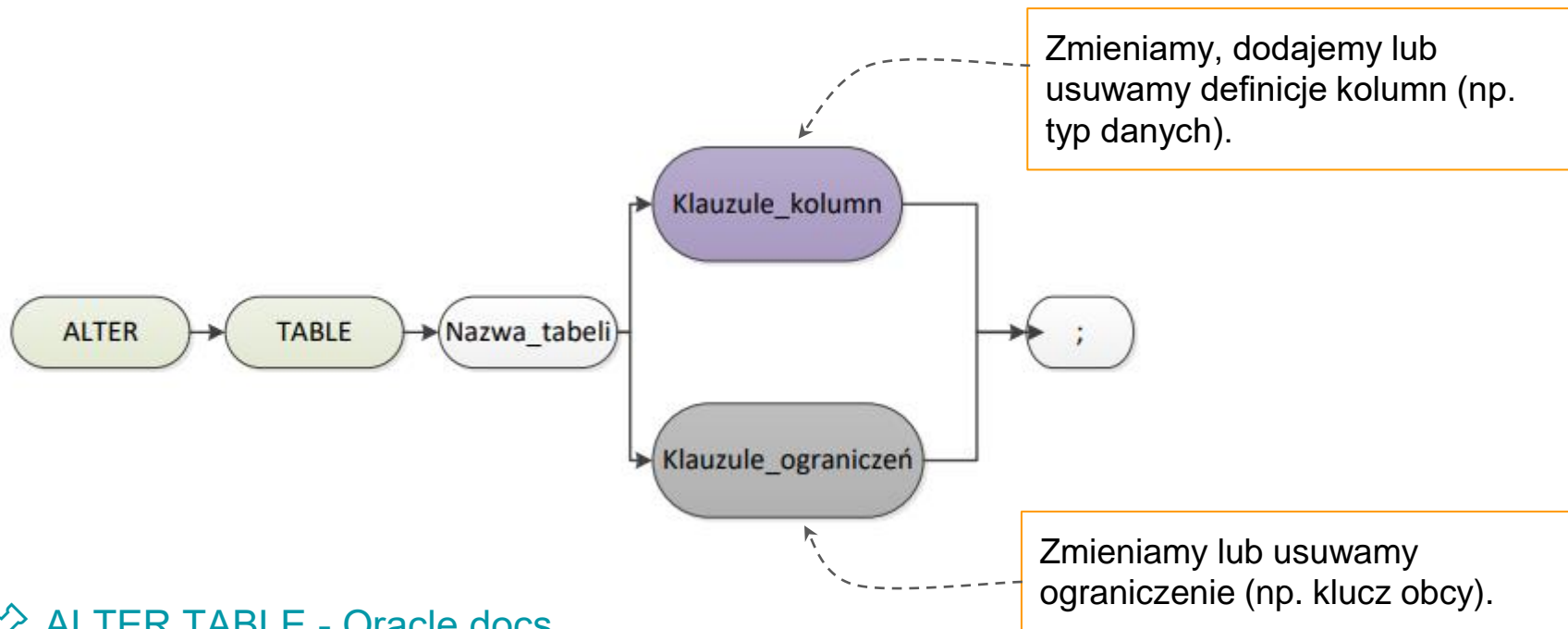
Sprawdźcie skróty klawiaturowe na waszych OS dla tych przycisków, pewnie Ctrl+Enter i F5.

Ćwiczenia - polecenie CREATE

1. Z wykorzystaniem polecenia `CREATE TABLE` utwórz tabelę `STANOWISKA`, która będzie przechowywać podstawowe dane o stanowiskach np. identyfikator, minimalną i maksymalną płacę, nazwę stanowiska.
2. Z wykorzystaniem polecenia `CREATE TABLE` utwórz tabele `REGIONY` oraz `KRAJE`. Pamiętaj o umieszczeniu klucza obcego w ramach tabeli `KRAJE` do tabeli `REGIONY`.

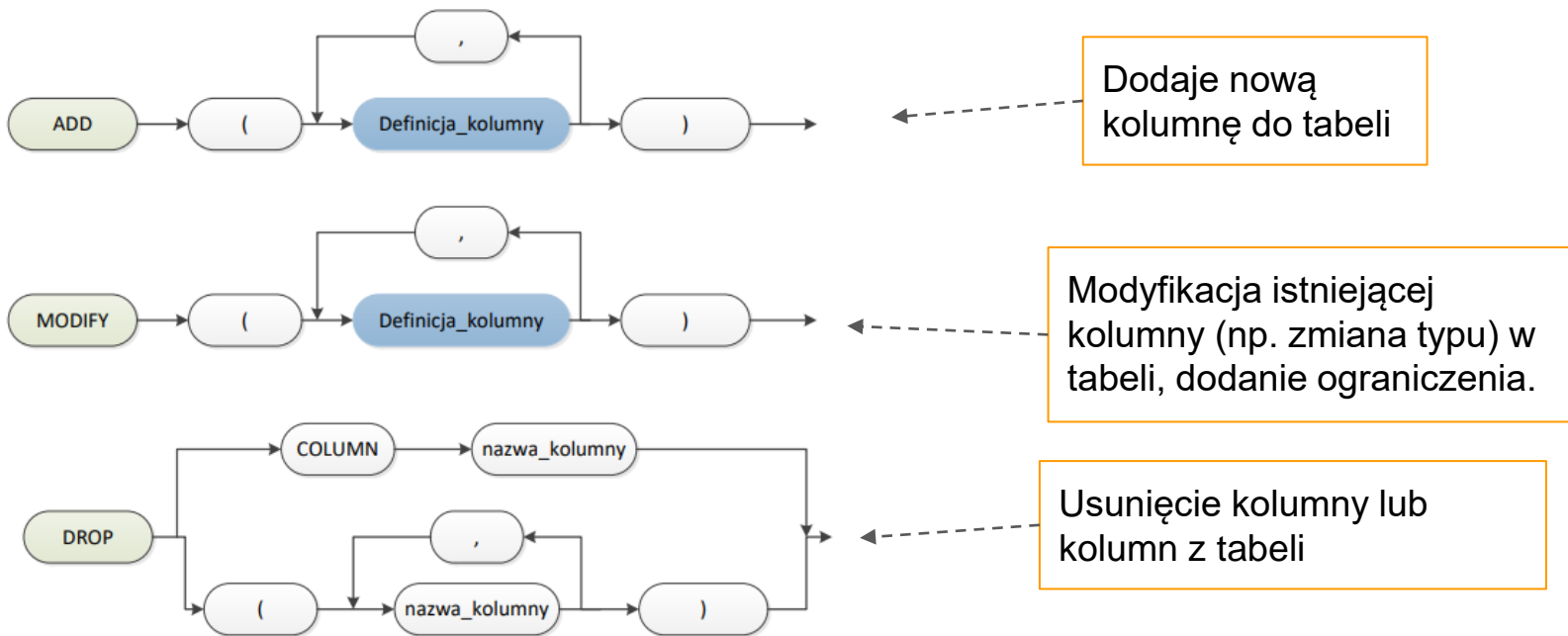
Polecenie ALTER TABLE

Podstawowy schemat polecenia jest następujący:



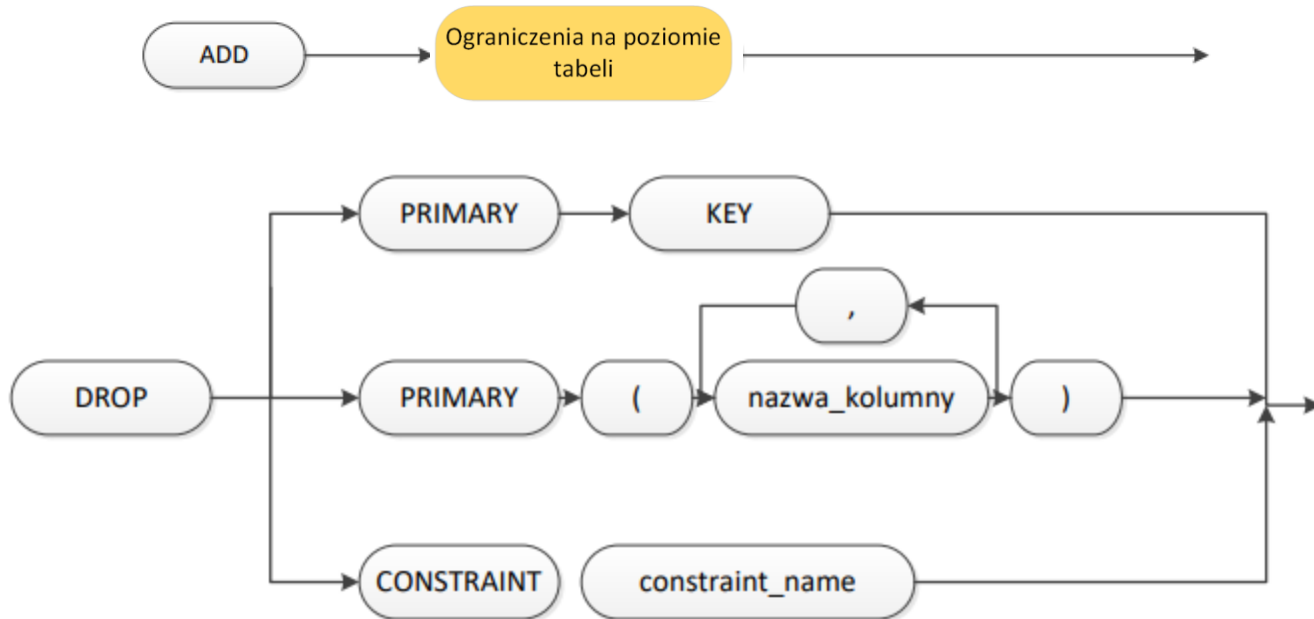
Polecenie ALTER TABLE (cd.)

Gdzie **klauszule_kolumn** można zdefiniować następująco:



Polecenie ALTER TABLE (cd.)

Gdzie `klauzule_ograniczeń` można zdefiniować następująco (`ograniczenia na poziomie tabeli` definiowane są tak jak w przypadku polecenia CREATE TABLE):



Alternatywny sposób definiowania klucza głównego za pomocą polecenia ALTER TABLE

```
CREATE TABLE pracownicy
(
    id_pracownika      NUMBER (4) NOT NULL
    , imie              VARCHAR2 (40 byte) NOT NULL
    , nazwisko          VARCHAR2 (40 byte) NOT NULL
    , zarobki           NUMBER (7, 2)
    , data_zatrudnienia DATE
    , id_zakladu        NUMBER (4) NOT NULL
    , id_kierownika     NUMBER (4)
    , kod_stanowiska    NUMBER (4) NOT NULL
);

ALTER TABLE pracownicy
ADD CONSTRAINT pracownicy_pk PRIMARY KEY ( id_pracownika );
```

Definiowanie klucza obcego

```
ALTER TABLE pracownicy
  ADD CONSTRAINT pracownicy_stanowiska_fk FOREIGN KEY ( kod_stanowiska )
  REFERENCES stanowiska ( kod_stanowiska );
```

Nazwa ograniczenia

Kolumna będąca kluczem obcym

Odwołanie do klucza głównego z innej tabeli

-- drugi sposób (podczas definiowania tabeli Pracownicy)

```
CREATE TABLE pracownicy
(
  id_pracownika      NUMBER (4) NOT NULL CONSTRAINT pracownicy_pk PRIMARY KEY
  , imie              VARCHAR2 (40 byte) NOT NULL
  , nazwisko          VARCHAR2 (40 byte) NOT NULL
  , zarobki           NUMBER (7, 2)
  , data_zatrudnienia DATE
  , id_zakladu         NUMBER (4) NOT NULL
  , id_kierownika     NUMBER (4)
  , kod_stanowiska    NUMBER (4) NOT NULL REFERENCES stanowiska (
    kod_stanowiska )
);
```

Ćwiczenia - polecenie ALTER TABLE

1. Za pomocą polecenia CREATE TABLE utwórz tabelę ZAKŁADY, w której przechowywane są dane o zakładach pracy. Następnie poleceniem ALTER TABLE dodaj klucz obcy do tabeli ZAKŁADY, który wskaże managera danego zakładu.
2. Dla tabeli STANOWISKA z użyciem polecenia ALTER TABLE ogranicz płace na stanowiskach do min. 1000.

Dodatkowe polecenia (DDL)

Zmiana nazwy tabeli

➤ `RENAME old_name TO new_name;`

Usunięcie tabeli:

➤ `DROP TABLE table [CASCADE CONSTRAINTS | PURGE]`

Przywrócenie tabeli:

➤ `FLASHBACK TABLE table TO BEFORE DROP;`



[RENAME - Oracle docs](#)



[DROP - Oracle docs](#)



[FLASHBACK - Oracle docs](#)



Praca domowa

1. Wykonaj wszystkie polecenia z zajęć.
2. Stwórz tabelę Szkolenia zawierającą kolumnę identyfikator i nazwa.
3. Zmodyfikuj tabelę Pracownicy dodając kolumnę szkolenie_id będące kluczem obcym do tabeli Szkolenia.
4. Zmodyfikuj tabelę Pracownicy usuwając kolumnę date_zatrudnienia.
5. Usuń tabelę Zakłady. Co obserwujesz? Dlaczego? Jak wymusić usunięcie tabeli?
6. Przywróć tabelę Zakłady poleceniem `FLASHBACK tablename TO BEFORE DROP`.
7. Co zmienia klauzula `PURGE` przy `DROP table`? Wypróbuj na tabeli Adresy. Jak wtedy odtworzyć dane?