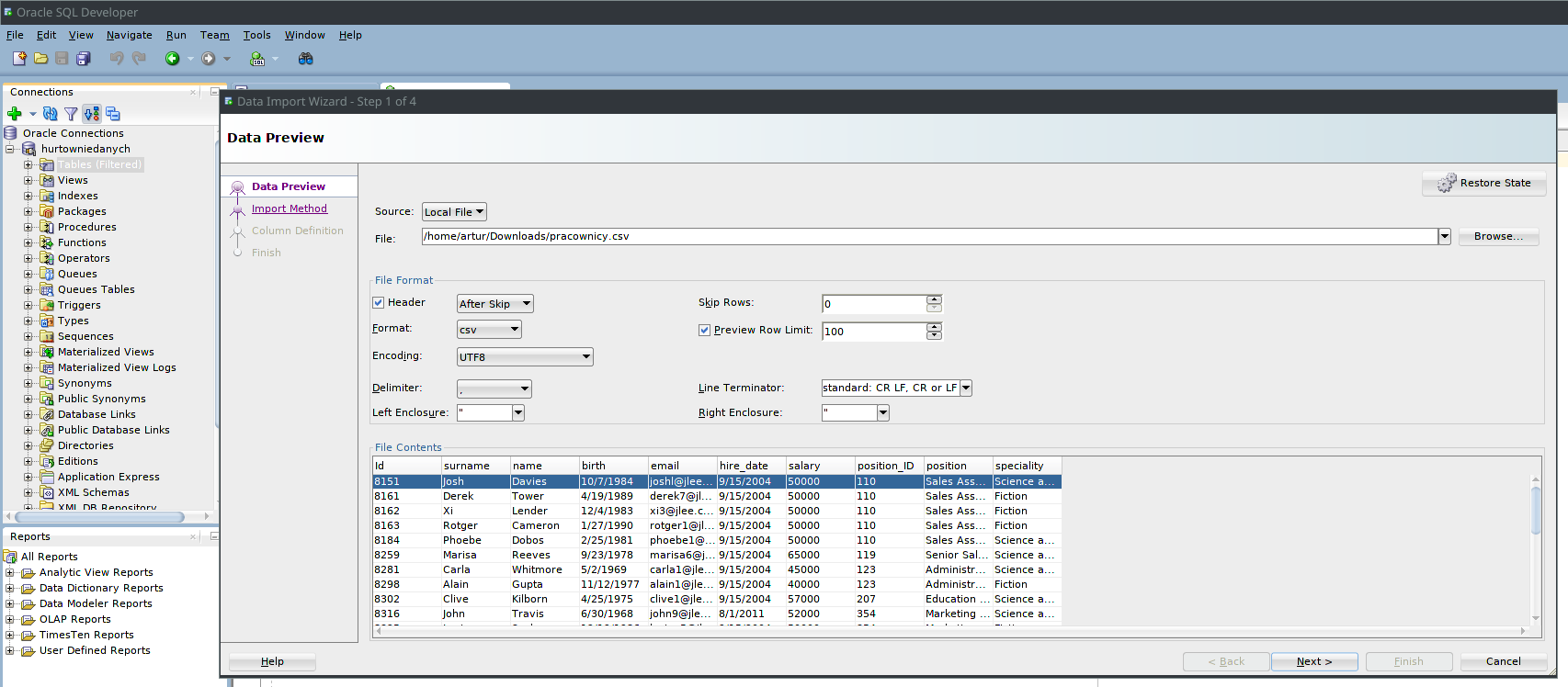
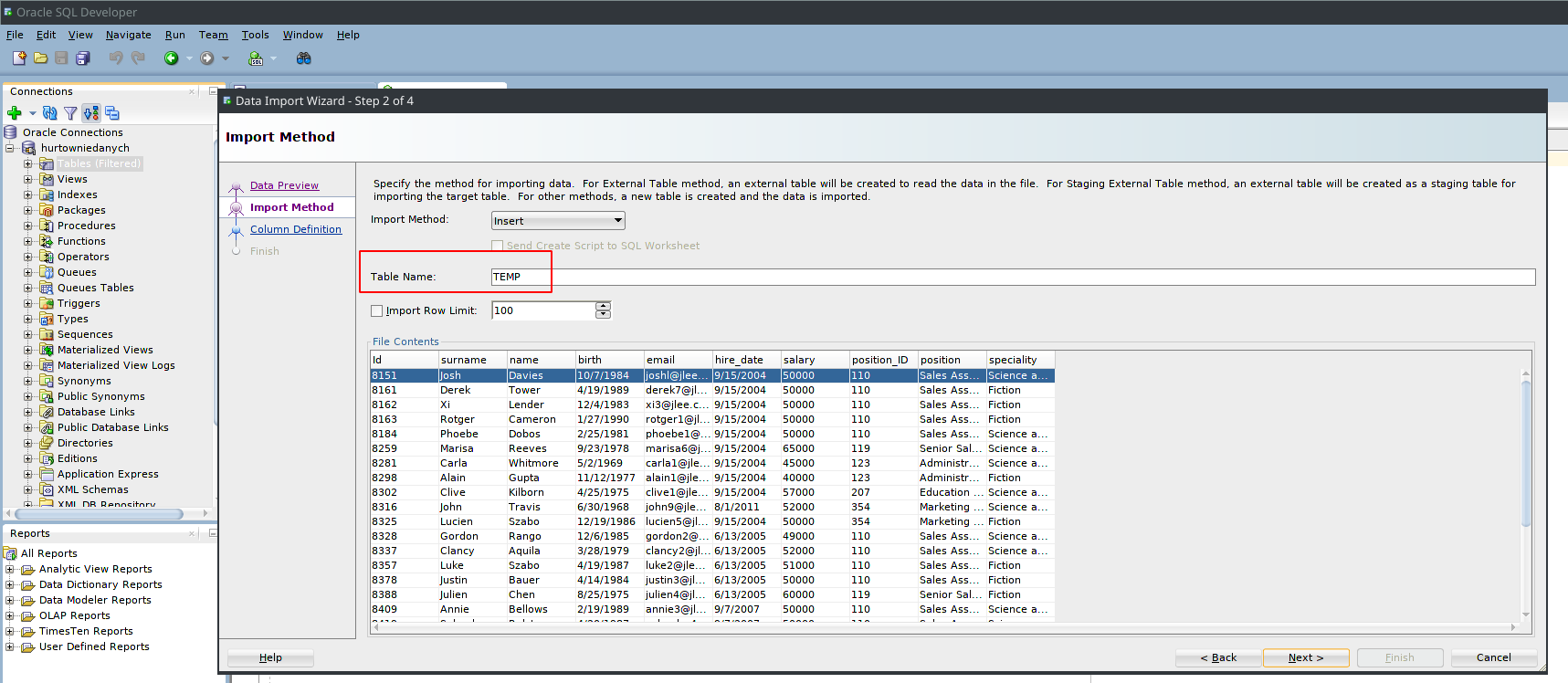
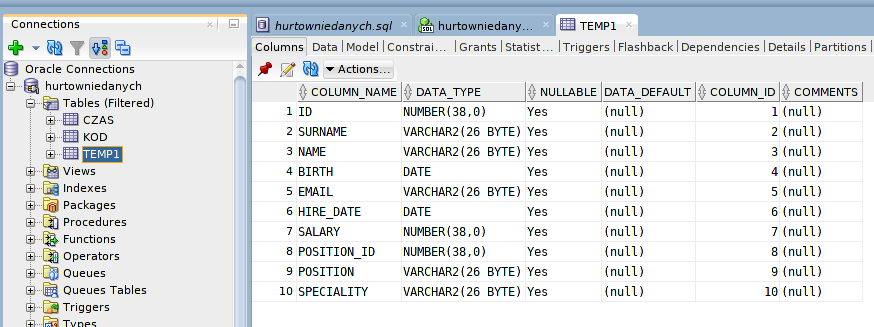
# Zadanie 1







# Zadanie 2

CREATE TABLE stanowisko (

id\_stanowiska NUMBER PRIMARY KEY,

nazwa VARCHAR(50)

);

CREATE TABLE specjalnosc (

id\_specjalnosci NUMBER PRIMARY KEY,

nazwa VARCHAR(50)

);

CREATE TABLE pracownik (

id\_pracownika NUMBER PRIMARY KEY,

surname VARCHAR(50),

name VARCHAR(50),

birth DATE NOT NULL,

hire DATE NOT NULL,

email VARCHAR(50),

salary NUMBER NOT NULL,

id\_stanowiska

REFERENCES stanowisko ( id\_stanowiska ),

id\_specjalnosci

REFERENCES specjalnosc ( id\_specjalnosci )

);

# 

# Zadanie 3

CREATE SEQUENCE sek\_stanowisko;

/

CREATE SEQUENCE sek\_specjalnosc;

/

CREATE SEQUENCE sek\_pracownik;

/

CREATE OR REPLACE TRIGGER t\_ustaw\_id\_stanowisko BEFORE

INSERT ON stanowisko

FOR EACH ROW

BEGIN

:new.id\_stanowiska := sek\_stanowisko.nextval;

END;

/

CREATE OR REPLACE TRIGGER t\_ustaw\_id\_specjalnosc BEFORE

INSERT ON specjalnosc

FOR EACH ROW

BEGIN

:new.id\_specjalnosci := sek\_specjalnosc.nextval;

END;

/

CREATE OR REPLACE TRIGGER t\_ustaw\_id\_pracownik BEFORE

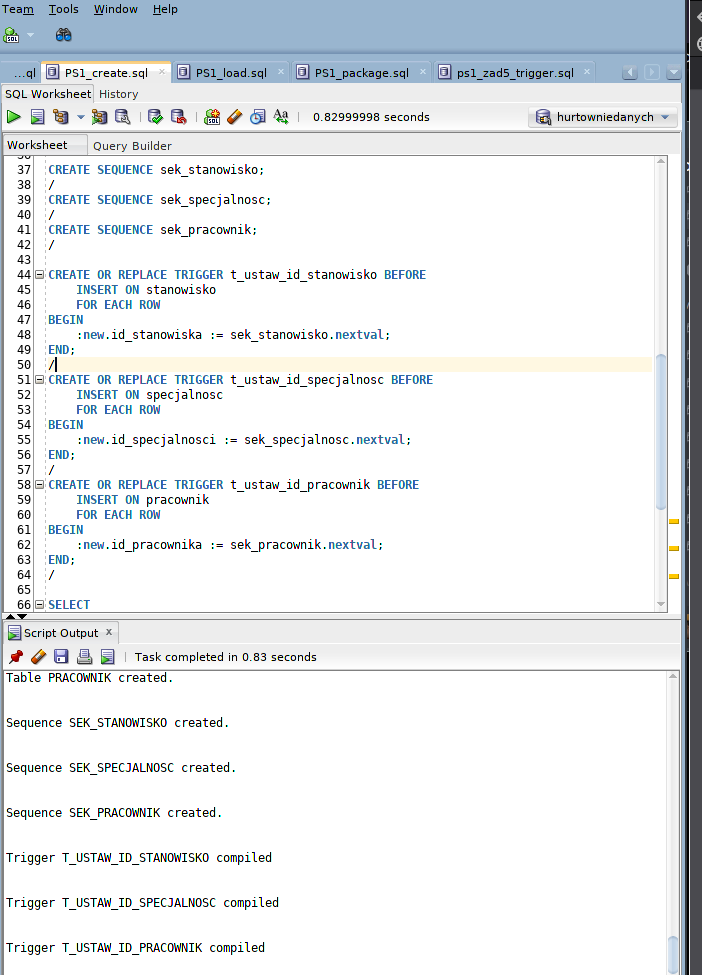
INSERT ON pracownik

FOR EACH ROW

BEGIN

:new.id\_pracownika := sek\_pracownik.nextval;

END;



# Zadanie 4

CREATE OR REPLACE PACKAGE zad4oraz5 AUTHID definer AS

PROCEDURE reshape\_data;

FUNCTION avg\_salary (

nazwa\_stanowiska VARCHAR2

) RETURN NUMBER;

END zad4oraz5;

/

CREATE OR REPLACE PACKAGE BODY zad4oraz5 AS

PROCEDURE reshape\_data IS

BEGIN

INSERT INTO specjalnosc ( nazwa )

SELECT DISTINCT

speciality AS name

FROM

temp1;

INSERT INTO stanowisko ( nazwa )

SELECT DISTINCT

position AS nazwa

FROM

temp1;

INSERT INTO pracownik (

surname,

name,

birth,

email,

hire,

salary,

id\_stanowiska,

id\_specjalnosci

)

SELECT

temp1.surname,

temp1.name,

temp1.birth,

temp1.email,

temp1.hire\_date,

temp1.salary,

stanowisko.id\_stanowiska,

specjalnosc.id\_specjalnosci

FROM

temp1

INNER JOIN stanowisko ON temp1.position = stanowisko.nazwa

INNER JOIN specjalnosc ON temp1.speciality = specjalnosc.nazwa;

END reshape\_data;

FUNCTION avg\_salary (

nazwa\_stanowiska VARCHAR2

) RETURN NUMBER IS

srednia\_pensja NUMBER;

BEGIN

SELECT

avgsal

INTO srednia\_pensja

FROM

(

SELECT

AVG(pracownik.salary) AS avgsal,

pracownik.id\_stanowiska AS idst

FROM

pracownik

GROUP BY

pracownik.id\_stanowiska

)

INNER JOIN stanowisko ON stanowisko.id\_stanowiska = idst

WHERE

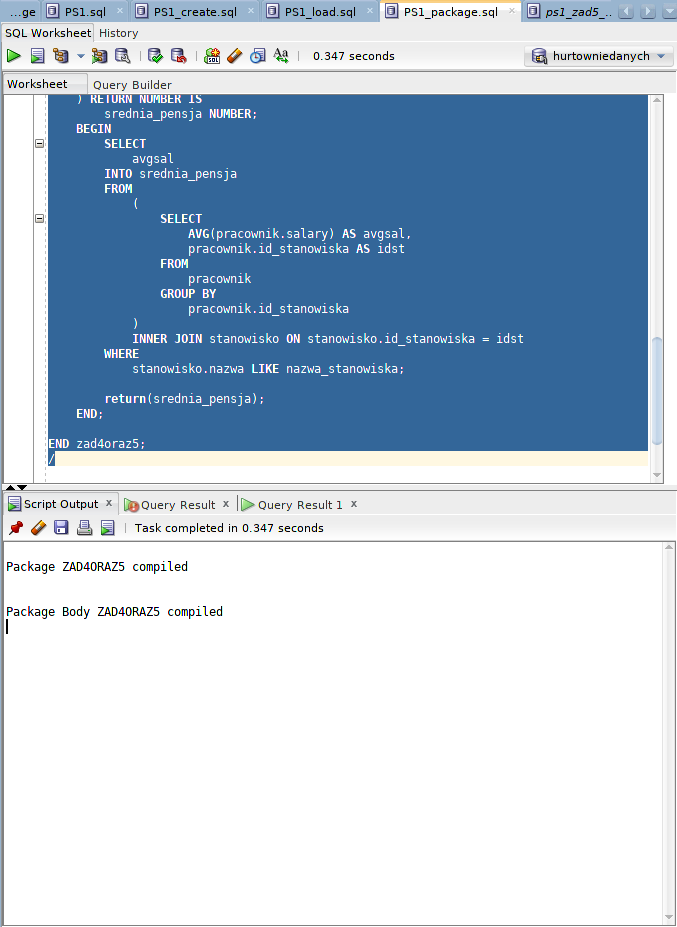
stanowisko.nazwa LIKE nazwa\_stanowiska;

**return**(srednia\_pensja);

END;

END zad4oraz5;

/



# Zadanie 5

CREATE OR REPLACE TRIGGER ogranicz\_pensje\_insert BEFORE

INSERT ON pracownik

FOR EACH ROW

DECLARE

avg\_salary\_result NUMBER;

BEGIN

SELECT

zad4oraz5.avg\_salary(st.nazwa)

INTO avg\_salary\_result

FROM

stanowisko st

WHERE

st.id\_stanowiska = :new.id\_stanowiska;

IF ( :new.salary > 1.3 \* avg\_salary\_result ) THEN

raise\_application\_error(-20001, ***'Pensja nie może być większa o 30% od średniej pensji na danym stanowisku'***);

END IF;

END;

/

CREATE OR REPLACE TRIGGER ogranicz\_pensje\_update BEFORE

UPDATE ON pracownik

FOR EACH ROW

DECLARE

avg\_salary\_result NUMBER;

BEGIN

SELECT

zad4oraz5.avg\_salary(st.nazwa)

INTO avg\_salary\_result

FROM

stanowisko st

WHERE

st.id\_stanowiska = :new.id\_stanowiska;

IF ( :new.salary > 1.3 \* avg\_salary\_result ) THEN

raise\_application\_error(-20001, ***'Pensja nie może być większa o 30% od średniej pensji na danym stanowisku'***);

END IF;

END;

INSERT INTO pracownik (

surname,

name,

birth,

hire,

email,

salary,

id\_stanowiska,

id\_specjalnosci

) VALUES (

***'Julien'***,

***'Chen '***,

TO\_DATE(***'25-AUG-75'***, ***'DD-MON-RR'***),

TO\_DATE(***'13-JUN-05'***, ***'DD-MON-RR'***),

***'julien4@jlee.com'***,

999999,

21,

21

);

UPDATE into pracownik

SET

surname = ***'Julien'***,

name = 'Chen',

birth = TO\_DATE(***'25-AUG-75'***, ***'DD-MON-RR'***),

hire = TO\_DATE(***'13-JUN-05'***, ***'DD-MON-RR'***),

email = ***'julien4@jlee.com'***,

salary = 999999,

id\_stanowiska = 21,

id\_specjalnosci = 21

WHERE

pracownik.surname LIKE ***'Julien'***

AND pracownik.name LIKE 'Chen'

AND pracownik.birth LIKE TO\_DATE(***'25-AUG-75'***, ***'DD-MON-RR'***);

/