

SQL

Structured Query Language

Язык Структурированных Запросов



Кому необходимо знать SQL?

DBA

Reporting specialist

Программист

Тестировщик

Бизнес
аналитик

И многим
другим

Для кого предназначен данный курс?

- Для людей, совсем ничего не знающих о SQL
- Для людей, которые хотят расширить знания о SQL или закрепить свои знания практическими заданиями
- Для людей, которые хотят подготовиться к экзамену 1Z0-071

Что Вы будете уметь по окончанию курса?

- Знать принципы работы SQL
- Уметь писать простые запросы
- Уметь писать сложные запросы
- Уметь создавать/изменять/удалять таблицы и множество других объектов
- Будете полностью готовы к сдаче экзамена 1Z0-071

```
SELECT * FROM employees  
WHERE department_id = 50 AND salary > 4000;
```

```
SELECT * FROM employees e  
WHERE manager_id IN (SELECT employee_id  
FROM employees WHERE  
TO_CHAR (hire_date, 'MM') = '01') AND  
(SELECT LENGTH (job_title) FROM jobs  
WHERE job_id = e.job_id) > 15;
```



Реализации SQL

Oracle
SQL

MySQL

PostgreSQL

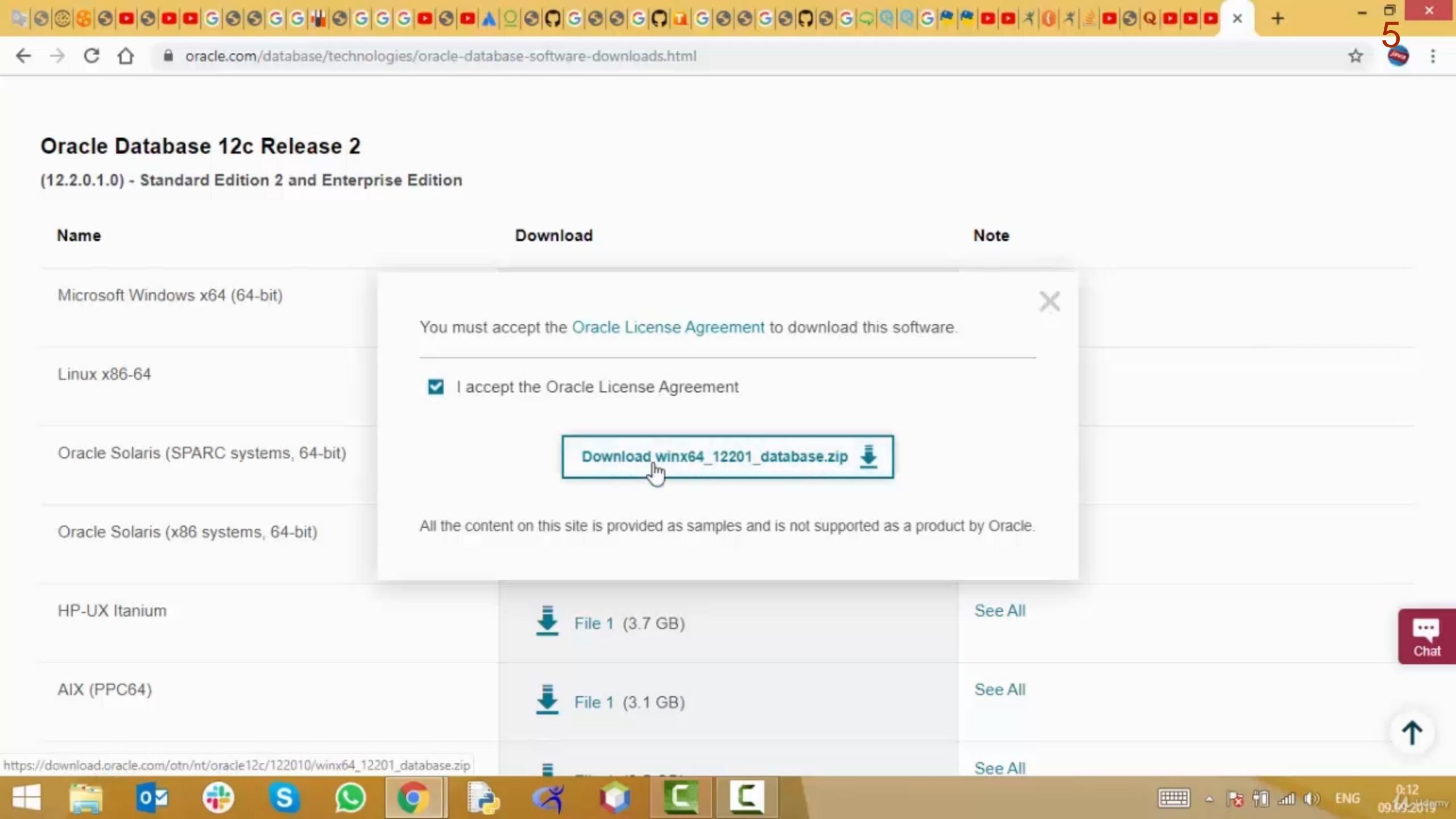
И другие

PL SQL – Procedure Language

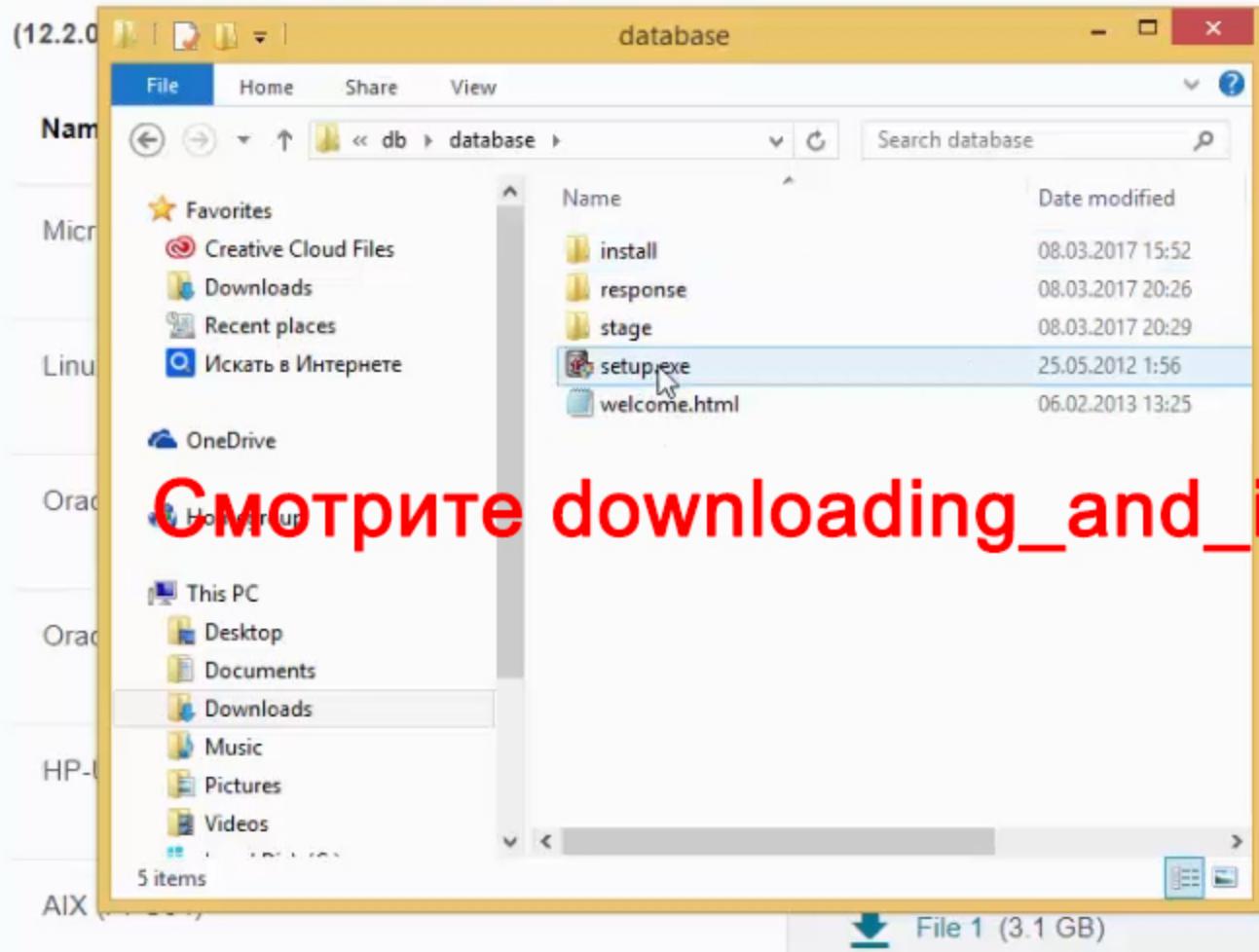
Для изучения SQL мы будем использовать

SQL Developer

sqlplus



Oracle Database 12c Release 2



Смотрите [downloading_and_installing_data_base.pdf](#)

Некоторые термины

- БД – совокупность данных, которые хранятся по определённым правилам и используются для удовлетворения информационных потребностей пользователей.
- Реляционная БД – это БД, где вся информация представляется в виде таблиц

students

<i>id</i>	<i>name</i>	<i>avg_grade</i>	<i>faculty_id</i>
1	Mike	8,2	1
2	John	9,1	3
3	Sara	7,5	1
4	Tim	6,8	2
5	Maria		

faculty

<i>id</i>	<i>name</i>	<i>dean</i>
1	Economics	James
2	CS	Amy
3	Philology	Luke

Некоторые термины

- БД – совокупность данных, которые хранятся по определённым правилам и используются для удовлетворения информационных потребностей пользователей.
- Реляционная БД – это БД, где вся информация представляется в виде таблиц

students

id	name	avg_grade	faculty_id
1	Mike	8,2	1
2	John	9,1	3
3	Sara	7,5	1
4	Tim	6,8	2
5	Maria		

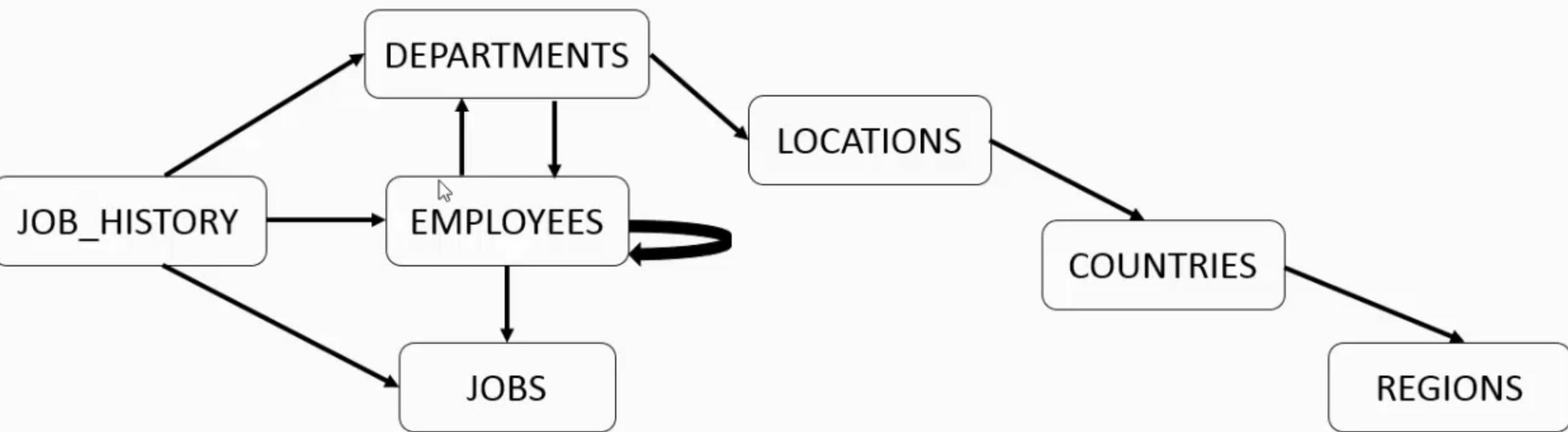
faculty

id	name	dean
1	Economics	James
2	CS	Amy
3	Philology	Luke

Знакомство с HR схемой

User – это лицо, которое может подключиться к БД (log on process).

DB схема – это все объекты в БД, которые принадлежат одному user-у.



SQL команды

DML

SELECT

INSERT

UPDATE

DELETE

MERGE

DDL

CREATE

ALTER

DROP

RENAME

TRUNCATE

TCL

COMMIT

ROLLBACK

SAVEPOINT

DCL

GRANT

REVOKE

Типы данных (data types)

Тип данных – это множество допустимых значений этих данных, а также совокупность операций над ними.

INTEGER

5

0

-200

NUMBER(p, s)

NUMBER(7, 2)

12345.67

123456.7

Где p – precision (общее максимальное количество цифр), s – scale (сколько из общего количества цифр отводится на дробную часть). Максимальное количество цифр на целую часть числа = p – s

Типы данных (data types)

CHAR(length)

CHAR(8)

privet

Privet, drug

Фиксированная длина

VARCHAR2(size)

VARCHAR2(8)

privet

Privet, drug

Переменная длина

Типы данных (data types)

DATE

11-Sep-19 23:17:18

Содержит информацию о году, месяце, дне, часе, минуте, секунде.

TIMESTAMP(f)

TIMESTAMP(6)

11-Sep-19 23:17:18.123000



Содержит информацию о году, месяце, дне, часе, минуте, секунде,олях секунды.

Понятие NULL

NULL – это отсутствие данных

0 или символ пробел – это не NULL. Они занимают место в памяти.

Результат арифметических операций с NULL – это всегда NULL.



Команда DESCRIBE

DESCRIBE SCHEMA.TABLE_NAME;

DESCRIBE hr.regions;

DESC hr.regions;

DESCRIBE regions;

DESC regions;

SELECT

3 фундаментальные концепции

PROJECTION – выбор столбцов из таблицы

SELECTION – выбор строк из таблицы

JOINING – объединение таблиц



SELECT statement (НЕ меняет данные)

Basic syntax

```
SELECT * FROM table;
```

```
SELECT * FROM countries;
```

```
SELECT column(s) FROM table;
```

```
SELECT country_name FROM countries;
```

```
SELECT country_id, country_name FROM countries;
```

SELECT statement (НЕ меняет данные)

DISTINCT

`SELECT DISTINCT column(s) FROM table;`

`SELECT DISTINCT job_id FROM job_history;`

`SELECT DISTINCT job_id, department_id
FROM job_history;`



File Edit View Navigate Run Source Team Tools Window Help

Connections

+ myconnection

Reports

Connections

myconnection

Oracle NoSQL Connections

Database Schema Service Connections

myconnection example1.sql

SQL Worksheet History

Worksheet Query Builder

```
select first_name, last_name, salary from employees
```

/

Query Result

Fetched 50 rows in 0,002 seconds

	FIRST_NAME	LAST_NAME	SALARY
1	Steven	King	24000
2	Neena	Kochhar	17000
3	Lex	De Haan	17000
4	Alexander	Hunold	9000
5	Bruce	Ernst	6000
6	David	Austin	4800
7	Valli	Pataballa	4800
8	Diana	Lorentz	4200
9	Nancy	Greenberg	12008
10	Daniel	Faviet	9000
11	Tiger	Gl	0000

Saved: myconnection~1 | Line 3 Column 52 | I



SELECT statement (НЕ меняет данные)

Expressions in select list

SELECT column(s), expression(s) FROM table;

SELECT salary*2 FROM employees;

SELECT first_name, salary, salary*1.5 FROM employees;

SELECT start_date, end_date, (end_date-start_date)+1
FROM job_history;



SELECT statement (НЕ меняет данные)

Alias

`SELECT column(s) alias, expression(s) alias FROM table;`

Alias – это альтернативное имя для столбца или целого выражения

`SELECT salary*2 AS bonus FROM employees;`

`SELECT start_date start, end_date end, (end_date - start_date)+1 "Count Of Days" FROM job_history;`

`SELECT 'First name is '||first_name||' and last_name is '||last_name our_employees FROM employees;`

SELECT statement (НЕ меняет данные)

Таблица DUAL

```
SELECT 'privet, '||'dorogoy student' AS greeting  
FROM dual;
```

```
SELECT 32*365*24*60*60 AS "moy vozrast v sekundax"  
FROM dual;
```

SELECT statement (НЕ меняет данные)

Проблемы с одинарными кавычками в тексте

```
SELECT 'It''s my life' AS song FROM dual;
```

Quote (q) operator:

q ' delimiter наш текст с кавычками delimiter '

```
SELECT q'<It's my life>' AS song FROM dual;
```

SELECT statement (НЕ меняет данные)

Итог первого знакомства с SELECT

```
SELECT * | {DISTINCT column(s) alias, expression(s) alias}  
FROM table;
```

I

SELECT

Концепция SELECTION

```
SELECT * | {DISTINCT column(s) alias, expression(s) alias}  
FROM table  
WHERE condition(s);
```

```
SELECT * FROM employees WHERE salary = 4800;
```

```
SELECT first_name, salary FROM employees  
WHERE last_name = 'King';
```

```
SELECT email FROM employees  
WHERE hire_date = '21.09.05';
```

SELECT

Операторы сравнения

=

>

<

>=

<=

!=

<>



SELECT

Операторы сравнения

=

>

<

>=

<=

!=

<>

I

BETWEEN

IN

SELECT

Операторы сравнения

= > < >= <= != <>

BETWEEN

IN

IS NULL

LIKE

%

_



SELECT

Логические операторы

Всё выражение принимает значение TRUE тогда и только тогда, когда все условия, объединённые **AND**, по отдельности тоже возвращают TRUE.

SELECT

Логические операторы

Всё выражение принимает значение TRUE тогда и только тогда, когда все условия, объединённые **AND**, по отдельности тоже возвращают TRUE.

Всё выражение принимает значение TRUE тогда и только тогда, когда хотя бы одно из условий, объединённых **OR**, по отдельности тоже возвращает TRUE.



SELECT

Логические операторы

Всё выражение принимает значение TRUE тогда и только тогда, когда все условия, объединённые **AND**, по отдельности тоже возвращают TRUE.

Всё выражение принимает значение TRUE тогда и только тогда, когда хотя бы одно из условий, объединённых **OR**, по отдельности тоже возвращает TRUE.



Оператор **NOT** меняет значение условия на противоположное.

SELECT

Последовательность выполнения операторов

ПРИОРИТЕТНОСТЬ	ОПЕРАТОР
1	()
2	/ * + -
4	
5	= > < >= <=
6	[NOT] LIKE, IS [NOT] NULL, [NOT] IN
7	[NOT] BETWEEN
8	!= <>
9	NOT
10	AND
11	OR

SELECT ORDER BY

```
SELECT * | {DISTINCT column(s) alias, expression(s) alias}  
FROM table  
WHERE condition(s)  
ORDER BY {col(s) | expr(s) | numeric I position}  
{ ASC | DESC } { NULLS FIRST | LAST };
```

```
SELECT first_name, salary FROM employees  
ORDER BY salary;
```

```
SELECT first_name, commission_pct FROM employees  
ORDER BY commission_pct DESC NULLS LAST;
```

SELECT ORDER BY

```
SELECT first_name, salary FROM employees  
ORDER BY 2;
```

```
SELECT job_id, first_name, last_name, salary, hire_date  
FROM employees  
ORDER BY job_id DESC, last_name, 4 DESC;
```

ФУНКЦИИ



Single-row



Multiple-row

Single-row функции



Character



Numeric



Date



Conversion



General

Character functions

Case conversion functions

- LOWER(s)



Character functions

Case conversion functions

- LOWER(s)
- UPPER(s)

Где s – это строка, текст.



Character functions

Case conversion functions

- LOWER(s)
- UPPER(s)
- INITCAP(s)

Где s – это строка, текст.



Character functions

Character manipulations functions

- CONCAT(s, s)

Где s – это строка, текст; n – конечная длина текста; p – текст для заполнения; trimstring – текст, который надо срезать;

Character functions

Character manipulations functions

- CONCAT(s, s)
- LENGTH(s)

Где s – это строка, текст; n – конечная длина текста; p – текст для заполнения; trimstring – текст, который надо срезать;

Character functions

Character manipulations functions

- CONCAT(s, s)
- LENGTH(s)
- LPAD(s, n, p)
- RPAD(s, n, p)



Где s – это строка, текст; n – конечная длина текста; p – текст для заполнения; trimstring – текст, который надо срезать;

Character functions

Character manipulations functions

- CONCAT(s, s)
- LENGTH(s)
- LPAD(s, n, p)
- RPAD(s, n, p)
- TRIM({trailing | leading | both} trimstring from s)

Где s – это строка, текст; n – конечная длина текста; p – текст для заполнения; trimstring – текст, который надо срезать;

Character functions

Character manipulations functions

- **INSTR(s, search string, start position, Nth occurrence)**

Где s – это строка, текст; search string – искомый текст; start position – позиция для начала работы; Nth occurrence – N-ое появление; number of characters – количество символов; search item – искомый элемент; replacement item – заменяющий элемент

Character functions

Character manipulations functions

- `INSTR(s, search string, start position, Nth occurrence)`
- `SUBSTR(s, start position, number of characters)`

Где s – это строка, текст; search string – искомый текст; start position – позиция для начала работы; Nth occurrence – N-ое появление; number of characters – количество символов; search item – искомый элемент; replacement item – заменяющий элемент



Character functions

Character manipulations functions

- `INSTR(s, search string, start position, Nth occurrence)`
- `SUBSTR(s, start position, number of characters)`
- `REPLACE(s, search item, replacement item)`

Где s – это строка, текст; search string – искомый текст; start position – позиция для начала работы; Nth occurrence – N-ое появление; number of characters – количество символов; search item – искомый элемент; replacement item – заменяющий элемент

Numemric functions

- ROUND(n, precision)

Где n – это число; precision – точность; dividend - делимое; divisor - делитель.

Numemric functions

- `ROUND(n, precision)`
- `TRUNC(n, precision)`



Где n – это число; precision – точность; dividend - делимое; divisor - делитель.

Numemric functions

- ROUND(n, precision)
- TRUNC(n, precision)
- MOD(dividend, divisor)

Где n – это число; precision – точность; dividend - делимое; divisor - делитель.



Date functions

- SYSDATE



Date functions

- SYSDATE
- MONTHS_BETWEEN(start_date, end_date)



Где start_date – дата «с»; end_date – дата «по»; date – дата; number_of_months - количество месяцев.

Date functions

- SYSDATE
- MONTHS_BETWEEN(start_date, end_date)
- ADD_MONTHS(date, number_of_months)

Где start_date – дата «с»; end_date – дата «по»; date – дата; number_of_months - количество месяцев.

Date functions

- `NEXT_DAY(date, day_of_the_week)`



Где date – дата; day of the week - день недели.

Date functions

- `NEXT_DAY(date, day_of_the_week)`
- `LAST_DAY(date)`



Где date – дата; day of the week - день недели.

Date functions

- **ROUND(date, date precision format)**

Где date – дата; date precision format – точность округления;

Век – CC; год – YYYY; четверть – Q; месяц – MM;
неделя – W; день – DD; час – HH; минута – MI.

Date functions

- `ROUND(date, date precision format)`
- `TRUNC(date, date precision format)`

Где date – дата; date precision format – точность округления;

Век – CC; год – YYYY; четверть – Q; месяц – MM;
неделя – W; день – DD; час – HH; минута – MI.

Conversion functions



TO_CHAR



TO_DATE



TO_NUMBER

TO_CHAR (number to char)

`TO_CHAR(number, format mask, nls_parameters) = T`

Конвертация числа в текст, используя функцию TO_CHAR означает взять число и сделать из него текст в том виде, в каком указан наш формат, если таковой имеется.

TO_CHAR (number to char)

Элемент	Описание	Формат	Число	Текст
9	Ширина	99999	18	18
0	Отображение нуля	099999	18	000018
.	Позиция десятичной точки	099999.999	18.35	000018.350
D	Позиция десятичного разделителя	099999D999	18.35	000018.350
,	Позиция запятой	099,999,999	1234567	001,234,567
G	Позиция разделителя групп	099G999G999	1234567	001,234,567
\$	Знак \$	\$099999	18	\$000018
L	Локальная валюта	L099999	18	\$000018
MI	Позиция знака -	099999MI	-18	000018-
PR	Скобки для отриц. чисел	099999PR	18	<000018>
S	Префикс + или -	S099999	18	+000018

TO_CHAR (date to char)

`TO_CHAR(date, format mask, nls_parameters) = T`

Конвертация даты в текст, используя функцию TO_CHAR означает взять дату и сделать из неё текст в том виде, в каком указан наш формат, если такой имеется.

TO_CHAR (date to char)

Дата для примера: '20-SEP-19'

Элемент	Описание	Текст
Y	Последняя цифра года	9 I
YY	Последние 2 цифры года	19
YYY	Последние 3 цифры года	019
YYYY	Год целиком	2019
RR	Год в формате 2-х цифр	19
YEAR	Буквенное написание года (Case-sensitive)	TWENTY NINETEEN
MM	Месяц в формате 2-х цифр	09
MON	3 буквы из названия месяца (Case-sensitive)	SEP
MONTH	Буквенное написание месяца (Case-sensitive)	SEPTEMBER

TO_CHAR (date to char)

Дата для примера: '20-SEP-19'

Элемент	Описание	Текст
D	День недели	6
DD	День месяца 2 цифры	20
DDD	День года	263
DY	3 буквы из назв. дня недели (Case-sensitive)	FRI
DAY	Полное название дня недели (Case-sensitive)	FRIDAY
W	Неделя месяца	3
WW	Неделя года	38
Q	Квартал года	3
CC	Век	21

TO_CHAR (date to char)

Дата для примера: '20-SEP-19 16:17:18'

Элемент	Описание	Текст
AM, PM, A.M. и P.M.	Индикатор	PM
HH, HH12 и HH24	Формат времени	04, 04, 16
MI	Минуты	17
SS	Секунды	18
SSSS	Секунды после полуночи	58638
- / ., ? # !	Пунктуация: 'MM.YY'	09.19
"Любой текст"	"Quarter" Q "of" Year'	Quarter 3 of Twenty Nineteen
TH	'DDth "of" Month'	20TH of September
SP	Буквенное написание(spell) 'MmSP Month Yyyysp'	Nine September Two Thousand Nineteen
THSP или SPTH	Комбинация: 'hh24SpTh'	sixteenth

TO_DATE (char to date)

`TO_DATE(text, format mask, nls_parameters) = D`

Конвертация текста в дату, используя функцию TO_DATE означает взять текст и объяснить в своём формате, где и как содержится информация о элементах даты в вашем тексте.



TO_NUMBER (char to number)

`TO_NUMBER(text, format mask, nls_parameters) = N`

Конвертация текста в число, используя функцию TO_NUMBER означает взять текст и объяснить в своём формате, где и как содержится информация о элементах числа в вашем тексте.

Connections Start Page myconnection.sql myconnection~1.sql myconnection~2.sql myconnection~3.sql myconnection~1 myconnection~2 myconnection~3

Worksheet Query Builder

```
select TO_DATE('18-09-87', 'DD-MM-RR'),  
       TO_CHAR(TO_DATE('18-09-87', 'DD-MM-RR'), 'DAY'),  
       LENGTH(TO_CHAR(TO_DATE('18-09-87', 'DD-MM-RR'), 'fmDAY'))  
  from dual;
```



```
select first_name, last_name, phone_number,  
       TO_NUMBER(SUBSTR(phone_number, INSTR(phone_number, '.') + 1), '999.9999') * 10000  
  form_num  
  from employees where employee_id < 130;
```

Query Result

All Rows Fetched: 30 in 0,004 seconds

FIRST_NAME	LAST_NAME	PHONE_NUMBER	FORM_NUMB
1 Steven	King	515.123.4567	1234567
2 Neena	Kochhar	515.123.4568	1234568
3 Lex	De Haan	515.123.4569	1234569
4 Alexander	Hunold	590.423.4567	4234567
5 Bruce	Ernst	590.423.4568	4234568
6 David	Austin	590.423.4569	4234569
7 Valli	Pataballa	590.423.4560	4234560
8 Diana	Lorentz	590.423.5567	4235567
9 Nancy	Greenberg	515.124.4569	1244569
10 Daniel	Faviet	515.124.4169	1244169
11 John	Chen	515.124.4269	1244269

General functions

- NVL(value, ifnull)



General functions

- NVL(value, ifnull)
- NVL2(value, ifnotnull, ifnull)



General functions

- NVL(value, ifnull)
- NVL2(value, ifnotnull, ifnull)
- NULLIF(value1, value2)



General functions

- NVL(value, ifnull)
- NVL2(value, ifnotnull, ifnull)
- NULLIF(value1, value2)
- COALESCE(value1, value2, ... , valueN)

Conditional functions

- `DECODE(expr, comp1, iftrue1, comp2, iftrue2, ..., compN, iftrueN, iffalse)`

Conditional functions

- `DECODE(expr, comp1, iftrue1, comp2, iftrue2, ..., compN, iftrueN, iffalse)`

- CASE

```
graph TD; CASE[• CASE] --> SimpleCASE[Simple CASE]; CASE --> SearchedCASE[Searched CASE]
```

Simple CASE

Searched CASE

Conditional function CASE

simple

searched

```
CASE expr  
WHEN comp1 THEN iftrue1  
WHEN comp2 THEN iftrue2  
.....  
WHEN compN THEN iftrueN  
ELSE iffalse  
END
```



Conditional function CASE

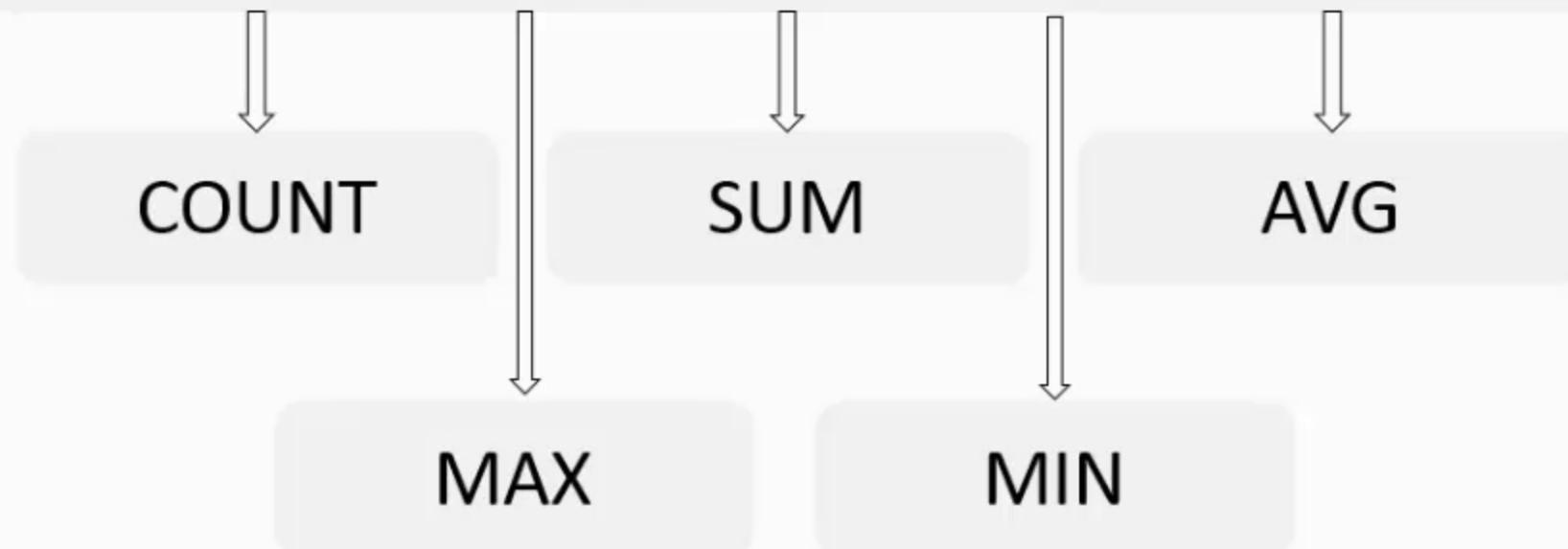
simple

```
CASE expr  
WHEN comp1 THEN iftrue1  
WHEN comp2 THEN iftrue2  
.....  
WHEN compN THEN iftrueN  
ELSE ifffalse  
END
```

searched

```
CASE  
WHEN cond1 THEN iftrue1  
WHEN cond2 THEN iftrue2  
.....  
WHEN condN THEN iftrueN  
ELSE ifffalse  
END
```

Group functions



Group functions

```
COUNT( { * | {DISTINCT | ALL}expression } )
```



Group functions

COUNT({ * | {DISTINCT | ALL}expression })

SUM({DISTINCT | ALL} expression)

Group functions

COUNT({ * | {DISTINCT | ALL}expression })

SUM({DISTINCT | ALL} expression)

AVG({DISTINCT | ALL} expression)

Group functions

COUNT({ * | {DISTINCT | ALL}expression })

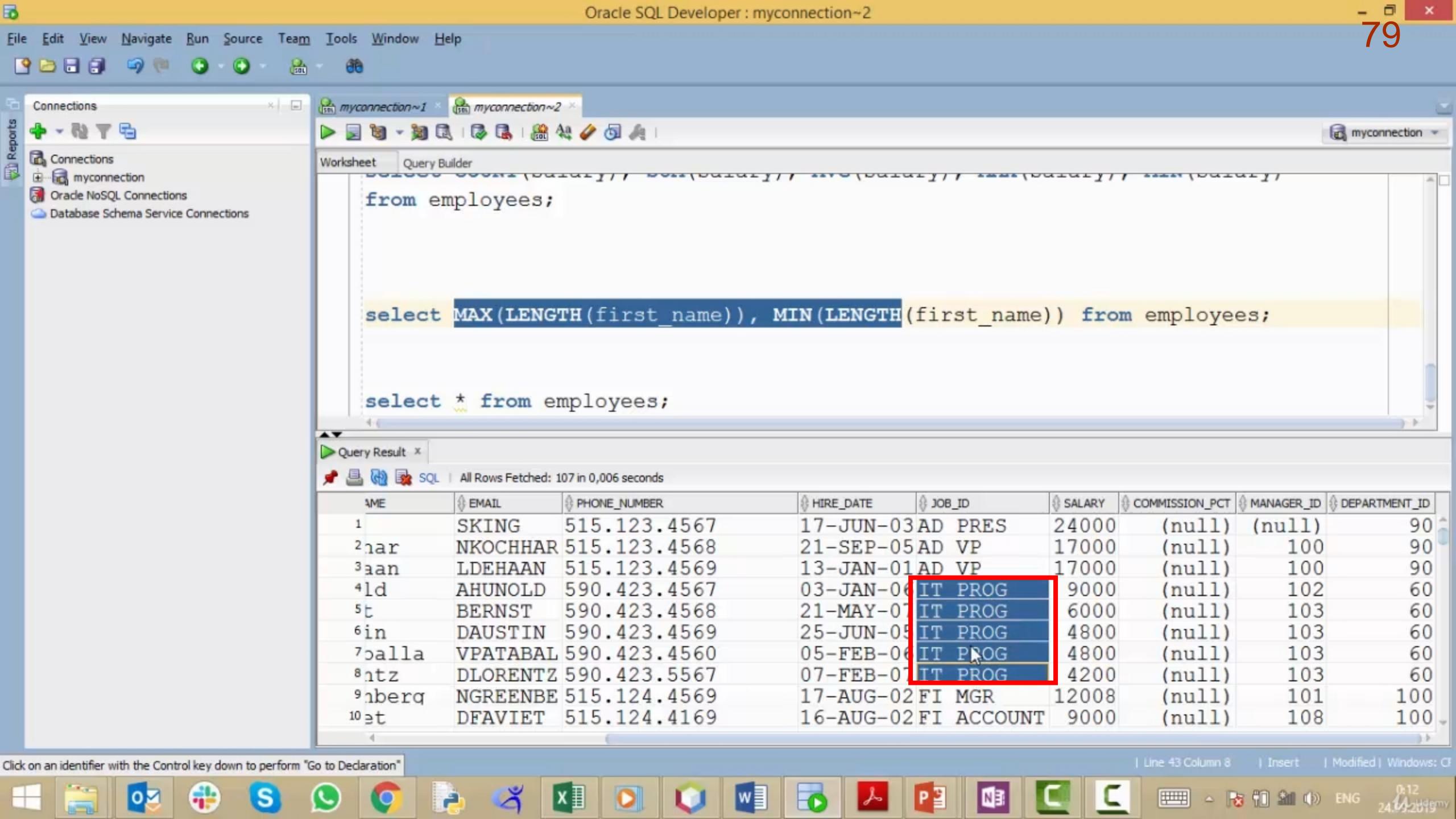
SUM({DISTINCT | ALL} expression)

AVG({DISTINCT | ALL} expression)

MIN({DISTINCT | ALL} expression)

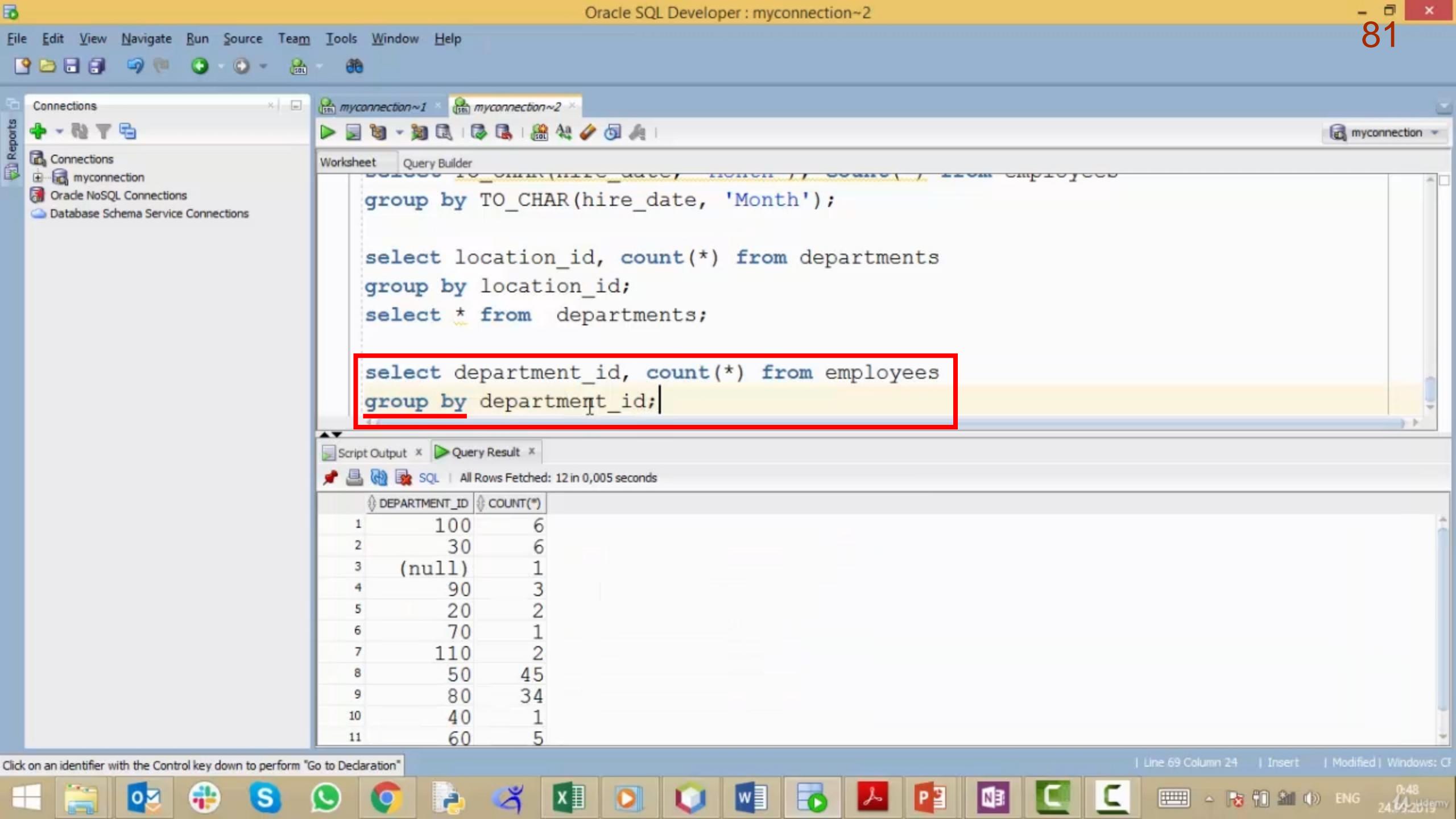
MAX({DISTINCT | ALL} expression)





GROUP BY

```
SELECT * | {DISTINCT column(s) alias, expression(s)  
alias, group_f-on(s)(col|expr alias), }  
FROM table  
WHERE condition(s)  
GROUP BY {col(s)|expr(s)}  
ORDER BY {col(s)|expr(s)|numeric position}  
{ ASC|DESC } { NULLS FIRST|LAST };
```



GROUP BY WITH HAVING

```
SELECT * | {DISTINCT column(s) alias, expression(s)
alias, group_f-on(s)(col|expr alias), }
FROM table
WHERE condition(s)
GROUP BY {col(s) | expr(s)}
HAVING group_condition(s)
ORDER BY {col(s) | expr(s) | numeric position}
{ ASC | DESC } { NULLS FIRST | LAST };
```

Connections myconnection~1 myconnection~2

Reports

Connections
myconnection
Oracle NoSQL Connections
Database Schema Service Connections

Worksheet Query Builder

```
select department_id, AVG(salary) from employees group by department_id;  
  
select SUM(AVG(salary)) from employees group by department_id;  
  
select MAX(SUM(AVG(salary))) from employees group by department_id;  
  
select ROUND(SUM(AVG(LENGTH(UPPER(last_name))))))  
from employees group by department_id;
```

Script Output Query Result

All Rows Fetched: 1 in 0,004 seconds

	ROUND(SUM(AVG(LENGTH(UPPER(LAST_NAME))))))
1	70

Click on an identifier with the Control key down to perform "Go to Declaration"

JOIN



INNER JOIN
(NATURAL JOIN)



OUTER JOIN



CROSS JOIN



JOIN

EQUIJOIN



NONEQUIJOIN

ORACLE JOIN

INNER JOIN

(NATURAL JOIN)



NATURAL
JOIN



USING



ON



NATURAL JOIN

```
SELECT column(s)  
FROM table_1  
NATURAL JOIN  
table_2;
```

```
SELECT *  
FROM regions  
NATURAL JOIN  
countries;
```

```
SELECT region_name,  
c.country_name, c.region_id  
FROM regions  
NATURAL JOIN  
countries c;
```

NATURAL JOIN with USING

```
SELECT column(s)
      FROM table_1
            JOIN
            table_2
      USING (column(s));
```

```
SELECT *
      FROM regions
            JOIN
            countries
      USING (region_id);
```

```
SELECT e.department_id,
           manager_id
      FROM employees e
            JOIN departments d
      USING (department_id);
```

NATURAL JOIN with ON

```
SELECT column(s)
      FROM table_1
            JOIN
              table_2
        ON (column1 = column2);
```

```
SELECT *
  FROM regions r
        JOIN
          countries c
    ON(r.region_id=c.region_id);
```

```
SELECT region_id
  FROM regions r
        JOIN
          countries c
    ON(r.region_id=c.region_id);
```

Connections

+ myconnection

Reports

Worksheet Query Builder

```
select * from countries;
select * from locations NATURAL JOIN countries NATURAL JOIN regions;

select * from locations JOIN countries USING (country_id)
JOIN regions USING (region_id);

select first_name, last_name, jh.job_id, start_date, end_date, department_name
from employees e JOIN job_history jh ON (e.employee_id = jh.employee_id)
JOIN departments d ON (jh.department_id = d.department_id);
```

Query Result

All Rows Fetched: 10 in 0,008 seconds

FIRST_NAME	LAST_NAME	JOB_ID	START_DATE	END_DATE	DEPARTMENT_N...
Neena	Kochhar	AC ACCOUNT	21-SEP-97	27-OCT-01	Accounting
Neena	Kochhar	AC MGR	28-OCT-01	15-MAR-05	Accounting
Lex	De Haan	IT PROG	13-JAN-01	24-JUL-06	IT
Den	Raphaely	ST CLERK	24-MAR-06	31-DEC-07	Shipping
Payam	Kaufling	ST CLERK	01-JAN-07	31-DEC-07	Shipping
Jonathon	Taylor	SA REP	24-MAR-06	31-DEC-06	Sales
Jonathon	Taylor	SA MAN	01-JAN-07	31-DEC-07	Sales
Jennifer	Whalen	AD ASST	17-SEP-95	17-JUN-01	Executive
Jennifer	Whalen	AC ACCOUNT	01-JUL-02	31-DEC-06	Executive
Michael	Hartstein	MK REP	17-FEB-04	19-DEC-07	Marketing

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 11 Column 60



NONEQUIJOIN with ON

```
SELECT column(s)
      FROM table_1
            JOIN
              table_2
      ON (column1 {оператор неравенства} column2);
```



Connections



Connections

myconnection

Oracle NoSQL Connections

Database Schema Service Connections

...~3 myconnection~4 myconnection~5 myconnection~6 myconnection~7 myconnection~8 myconnection~9 myconnection~10



myconnection

Worksheet Query Builder

```
select employee_id, first_name, manager_id from employees;

select emp1.employee_id, emp1.first_name, emp1.manager_id
, emp2.first_name manager_name
from employees emp1 JOIN employees emp2 ON (emp1.manager_id = emp2.employee_id);
```

Query Result



SQL | Fetched 50 rows in 0,007 seconds

EMPLOYEE_ID	FIRST_NAME	MANAGER_ID	MANAGER_NAME
1	168 Lisa	148	Gerald
2	169 Harrison	148	Gerald
3	170 Tayler	148	Gerald
4	171 William	148	Gerald
5	172 Elizabeth	148	Gerald
6	173 Sundita	148	Gerald
7	103 Alexander	102	Lex
8	162 Clara	147	Alberto
9	163 Danielle	147	Alberto
10	164 Mattea	147	Alberto
11	165 David	147	Alberto

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 5 Column 81 | Insert | Modified | Windows: C



OUTER JOIN



LEFT OUTER
JOIN



RIGHT OUTER
JOIN



FULL OUTER
JOIN



OUTER JOIN

```
SELECT column(s) FROM  
table_1 LEFT OUTER JOIN table_2  
ON (column1 = column2);
```



OUTER JOIN

```
SELECT column(s) FROM  
table_1 LEFT OUTER JOIN table_2  
ON (column1 = column2);
```

```
SELECT column(s) FROM  
table_1 RIGHT OUTER JOIN table_2  
ON (column1 = column2);
```

OUTER JOIN

```
SELECT column(s) FROM  
table_1 LEFT OUTER JOIN table_2  
ON (column1 = column2);
```

```
SELECT column(s) FROM  
table_1 RIGHT OUTER JOIN table_2  
ON (column1 = column2);
```

```
SELECT column(s) FROM  
table_1 FULL OUTER JOIN table_2  
ON (column1 = column2);
```

CROSS JOIN

```
SELECT column(s)  
      I  
  FROM table_1  
 CROSS JOIN  
    table_2;
```

```
SELECT *  
FROM regions  
CROSS JOIN  
countries;
```

```
SELECT department_id,  
      I  
  FROM employees e  
 CROSS JOIN  
departments d;
```

ORACLE JOIN SYNTAX

IJ

```
SELECT column(s) FROM table_1 t1, table_2 t2  
      WHERE t1.column1 = t2.column2;
```

ROJ

```
SELECT column(s) FROM table_1 t1, table_2 t2  
      WHERE t1.column1 (+) = t2.column2I;
```

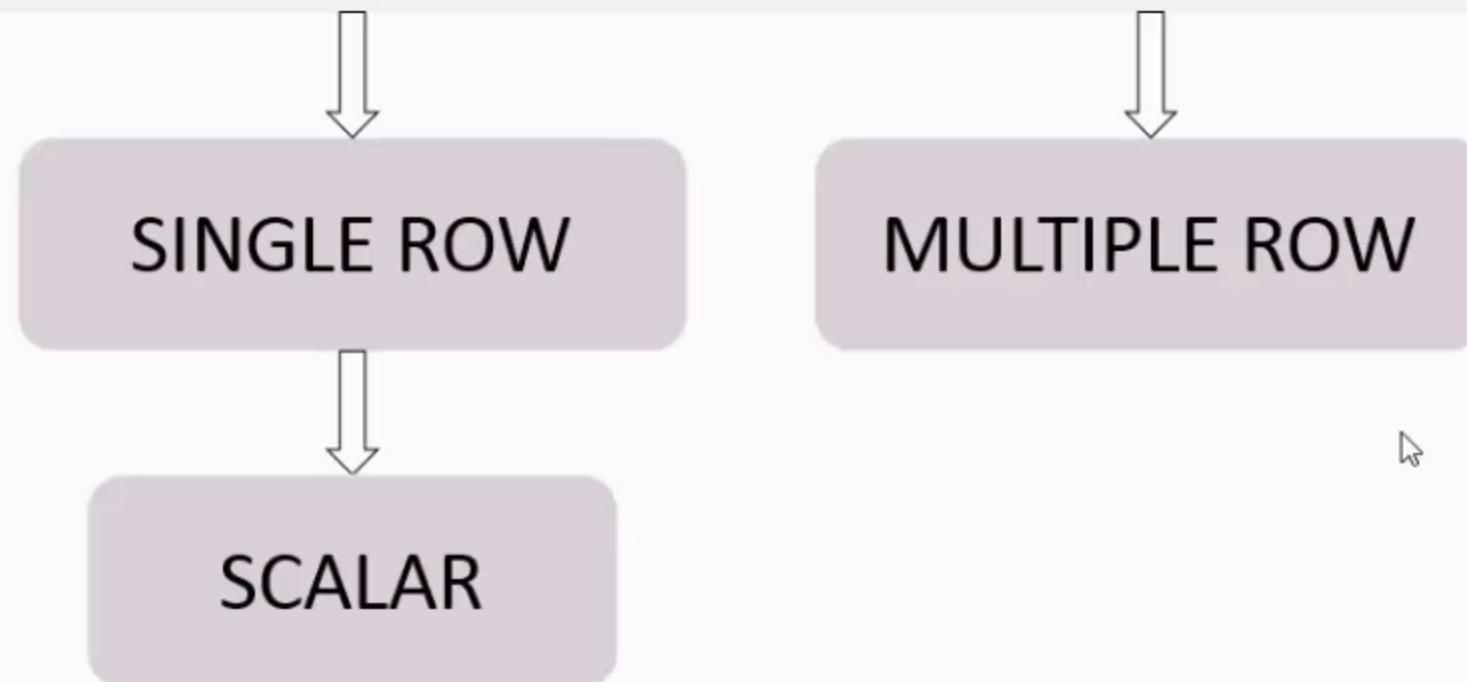
LOJ

```
SELECT column(s) FROM table_1 t1, table_2 t2  
      WHERE t1.column1 = t2.column2 (+);
```

CJ

```
SELECT column(s) FROM table_1 t1, table_2 t2;
```

TYPES OF SUBQUERIES



SINGLE ROW SUBQUERIES

```
SELECT first_name, last_name, salary  
      FROM employees  
     WHERE salary <  
          (SELECT MAX(salary)/5 FROM employees);
```

```
SELECT salary FROM employees  
     WHERE salary >=  
          (SELECT first_name, salary FROM employees  
             WHERE employee_id = 180);
```

```
SELECT salary FROM employees  
     WHERE salary =  
          (SELECT salary FROM employees  
             WHERE employee_id > 180);
```



MULTIPLE ROW SUBQUERIES

```
SELECT salary FROM employees  
      WHERE salary IN  
        (SELECT salary FROM employees  
          WHERE employee_id > 180);
```

```
SELECT first_name, last_name, salary FROM employees  
      WHERE salary > ANY (SELECT salary FROM employees  
          WHERE department_id = 100);
```

```
SELECT first_name, last_name, salary FROM employees  
      WHERE salary > ANY (SELECT salary, first_name FROM  
          employees WHERE department_id = 100);
```

CORRELATED SUBQUERIES

```
SELECT e1.first_name, e1.last_name, e1.salary  
      FROM employees e1  
     WHERE e1.salary >  
          (SELECT avg(e2.salary) FROM employees e2  
           WHERE e2.department_id=e1.department_id);
```

SET OPERATORS



UNION ALL

UNION ALL объединяет 2 аутпут множества в одно простым присоединением.

```
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'  
      UNION ALL  
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'
```

UNION

UNION объединяет 2 аутпут множества в одно, удаляя при этом дубликаты и сортируя его.

```
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'  
UNION  
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'
```



INTERSECT

INTERSECT 2-ух аутпут множеств возвращает только общие строки, удаляя при этом дубликаты и сортируя результат.

```
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'  
INTERSECT  
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'
```



MINUS

MINUS 2-ух аутпут множеств возвращает только те строки, которые есть в первом множестве, но нет во втором, удаляя при этом дубликаты и сортируя результат.

```
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'  
MINUS  
SELECT * FROM jobs  
WHERE job_id LIKE '%MAN%'
```



Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

...sql myconnection~18.sql × myconnection~1 × myconnection~2 × myconnection~3 × myconnection~4 × myconnection~5 × myconnection~6 × myconnection

Worksheet Query Builder

MINUS

```
select manager_id from employees where department_id = 40;

select department_id dep_id, to_char(null) job_id, sum(salary) from employees
group by department_id
UNION
select to_number(null), job_id, sum(salary) from employees
group by job_id;
```

Script Output × | Query Result × | Query Result 1 ×

All Rows Fetched: 31 in 0,005 seconds

DEP_ID	JOB_ID	SUM(SALARY)
7	70 (null)	10000
8	80 (null)	304500
9	90 (null)	58000
10	100 (null)	51608
11	110 (null)	20308
12	(null) AC ACCOUNT	8300
13	(null) AC MGR	12008
14	(null) AD ASST	4400
15	(null) AD PRES	24000
16	(null) AD VP	34000
17	(null) FI ACCOUNT	39600

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 90 Column 50 | Insert | Modified | Windows: CF



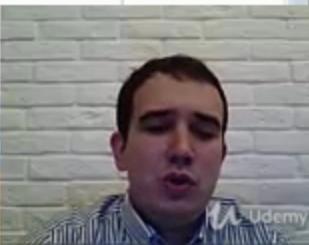
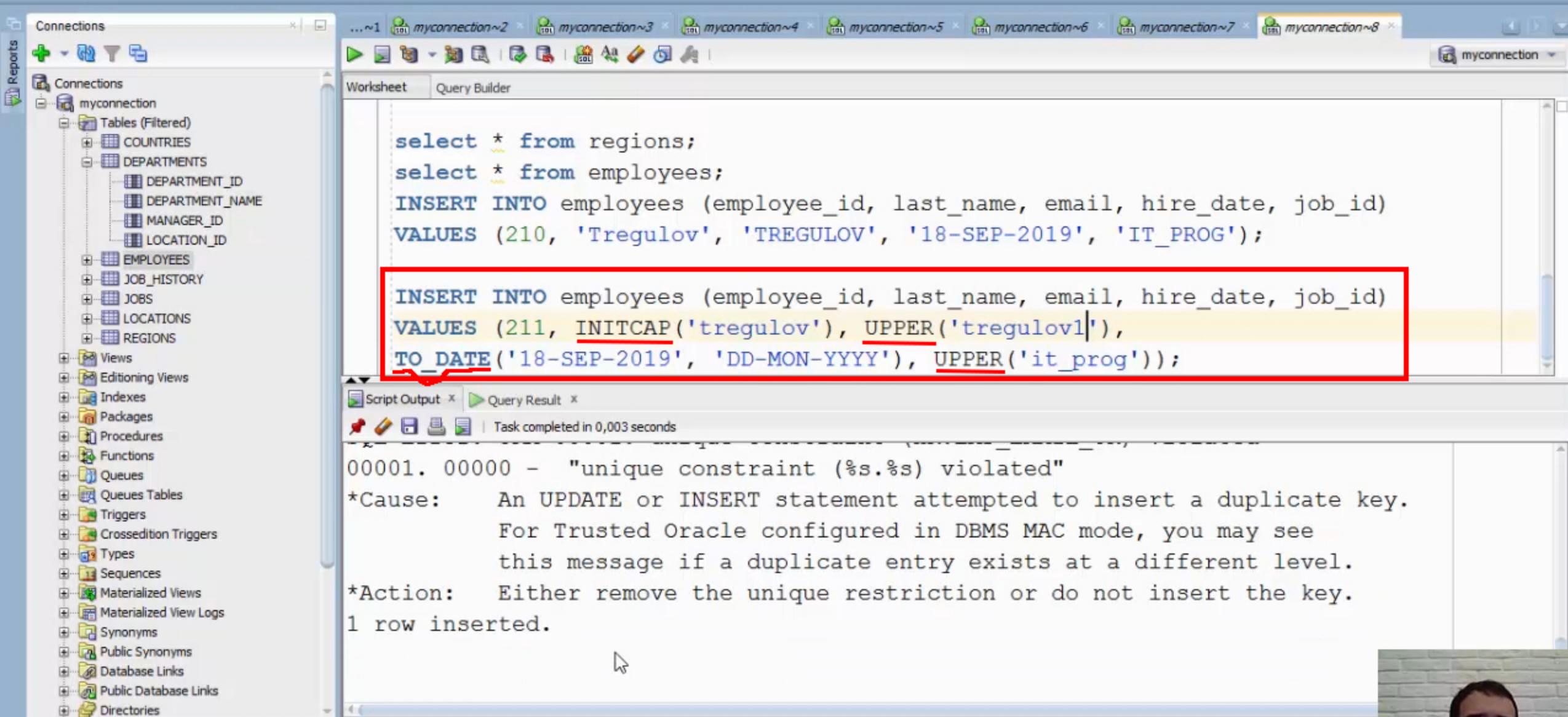
DML COMMANDS



INSERT

```
INSERT INTO table_name  
    column(s)  
VALUES (value(s));
```



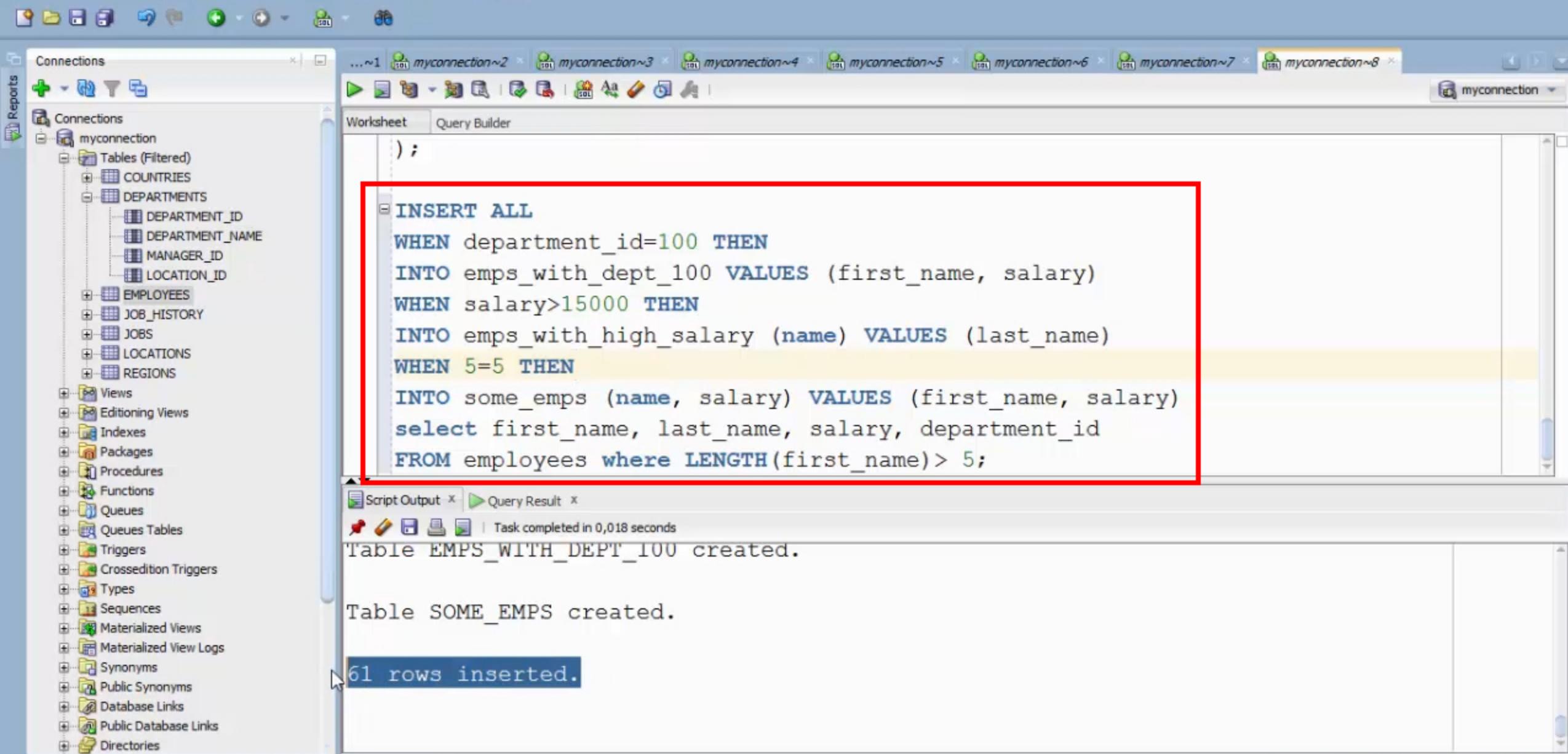


INSERT

```
INSERT INTO table_name  
    column(s)  
VALUES (value(s));
```



```
INSERT INTO table_name  
    column(s)  
SUBQUERY;
```



Click on an identifier with the Control key down to perform "Go to Declaration".

+ Страница



4 октября 2019 г.

1:18



Записная книжка... > Заметки на полях



(dept_id) sum(salary)

IVAN 18)

select DATE, INTEGER,
↓ SQL ошибка
select TIMESTAMP, NUMBERselect from
UNION
on

Taut Edge UNION



UPDATE

```
UPDATE table_name  
      SET  
column(s) = value(s)  
WHERE condition(s);
```



UPDATE

```
UPDATE table_name  
      SET  
    column(s) = value(s)  
 WHERE condition(s);
```

```
UPDATE table_name
```

```
      SET
```

```
    column(s) = subquery(s)
```

```
WHERE column = subquery;
```



DELETE

```
DELETE   
FROM table_name  
WHERE condition(s);
```

DELETE

```
DELETE  
FROM table_name  
WHERE condition(s);
```



```
DELETE  
FROM table_name  
WHERE column = subquery;
```

MERGE

```
MERGE INTO table_name1 t1
USING {table_name1|subquery} t2
ON (t1.column = t2.column)
WHEN MATCHED THEN
    UPDATE SET column=value
    DELETE WHERE condition
WHEN NOT MATCHED THEN
    INSERT(value1, value2)
VALUES (column1, column2);
```

ACID

ATOMICITY - АТОМАРНОСТЬ

CONSISTENCY - СОГЛАСОВАННОСТЬ

ISOLATION - ИЗОЛИРОВАННОСТЬ

DURABLE - ДОЛГОВЕЧНОСТЬ

TRANSACTIONS

5 октября 2019 г.

14:03

UPDATE t1

SELECT

DELETE t2

DELETE

INSERT t3

UPDATE

COMMIT / ROLLBACK

TRANSAC

TRANSAC
(ACID)

Update

Записная книжка... > Заметки на полях

Index

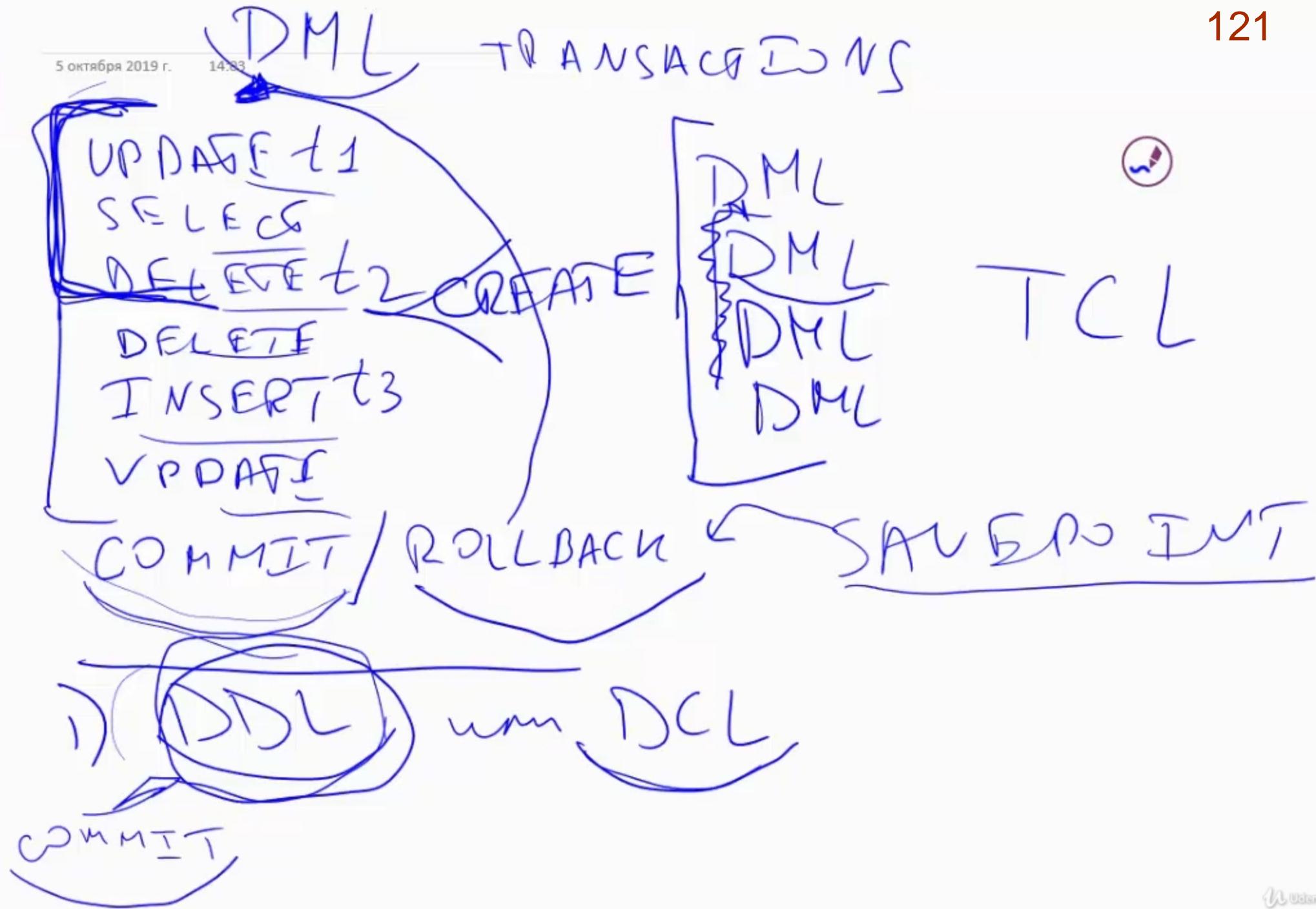
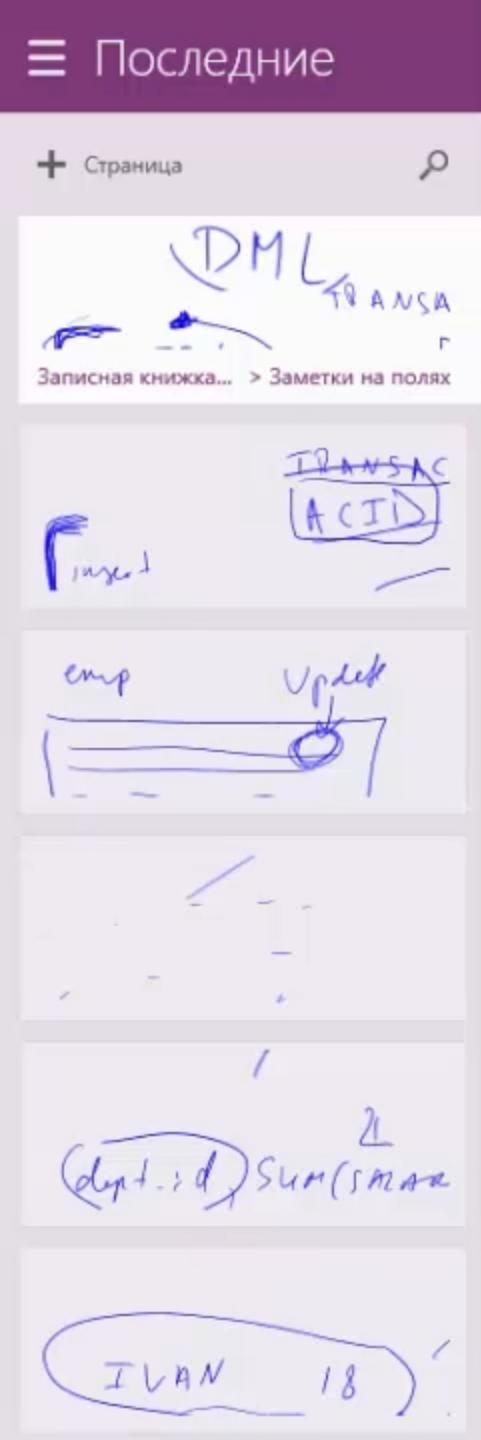
emp

1

(dept->d) sum(salary)

IVAN 18





DML
TRANSA

Записная книжка... > Заметки на полях

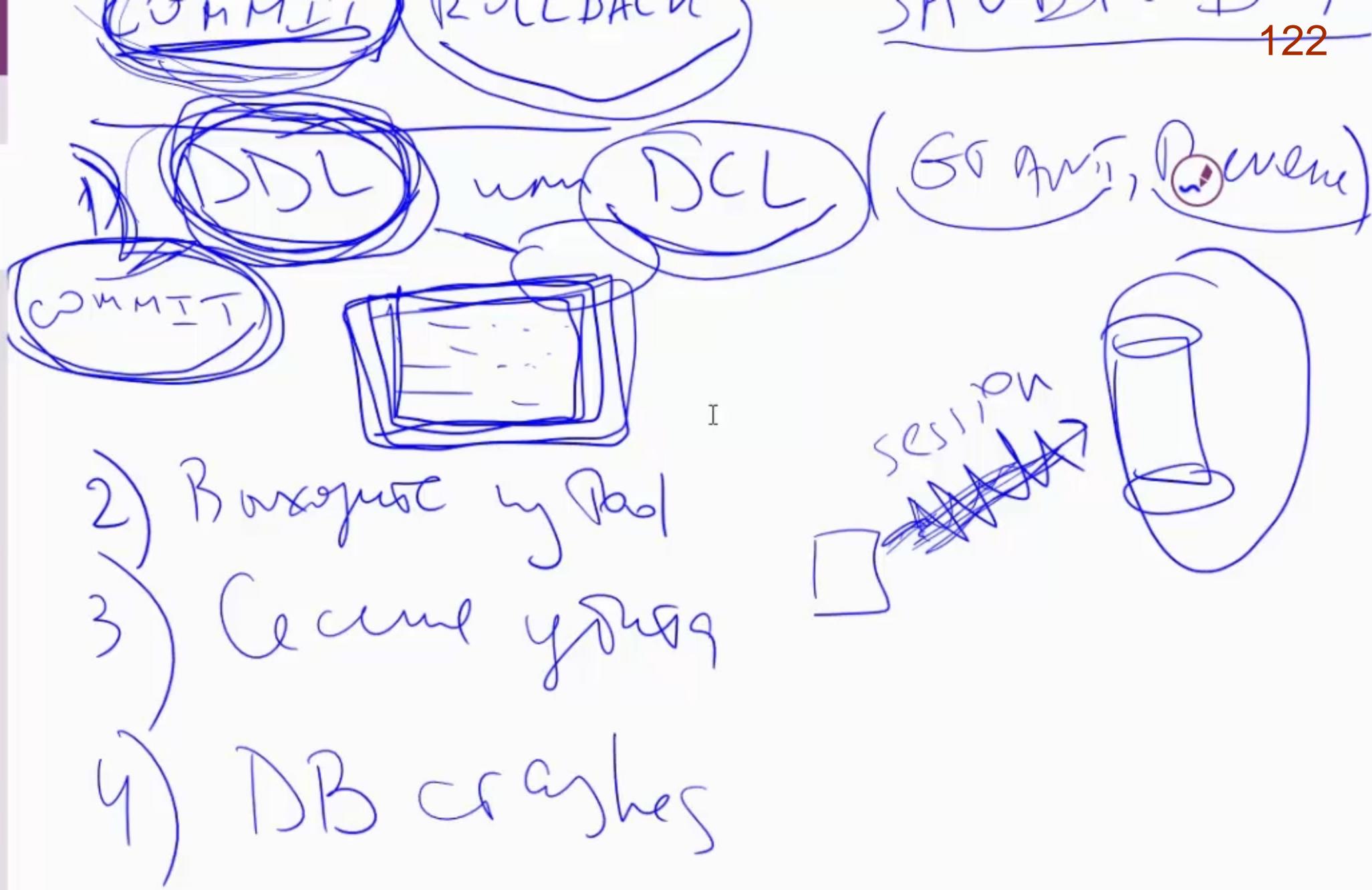
TRANSA
ACID

Trans

emp Update

(dept->d) sum(smax)

IVAN 18



DML
TRANS

Записная книжка... > Заметки на полях

Trans

TRANSA
C ID

emp Update

/

(dept->d) sum(smax)

DB crashes



DML
DML
DML

commit / rollback

SAVEPOINT

IVAN 18

TCL COMMANDS



COMMIT



TCL COMMANDS



COMMIT



ROLLBACK

ROLLBACK TO SAVEPOINT savepoint_name;



TCL COMMANDS



COMMIT



ROLLBACK

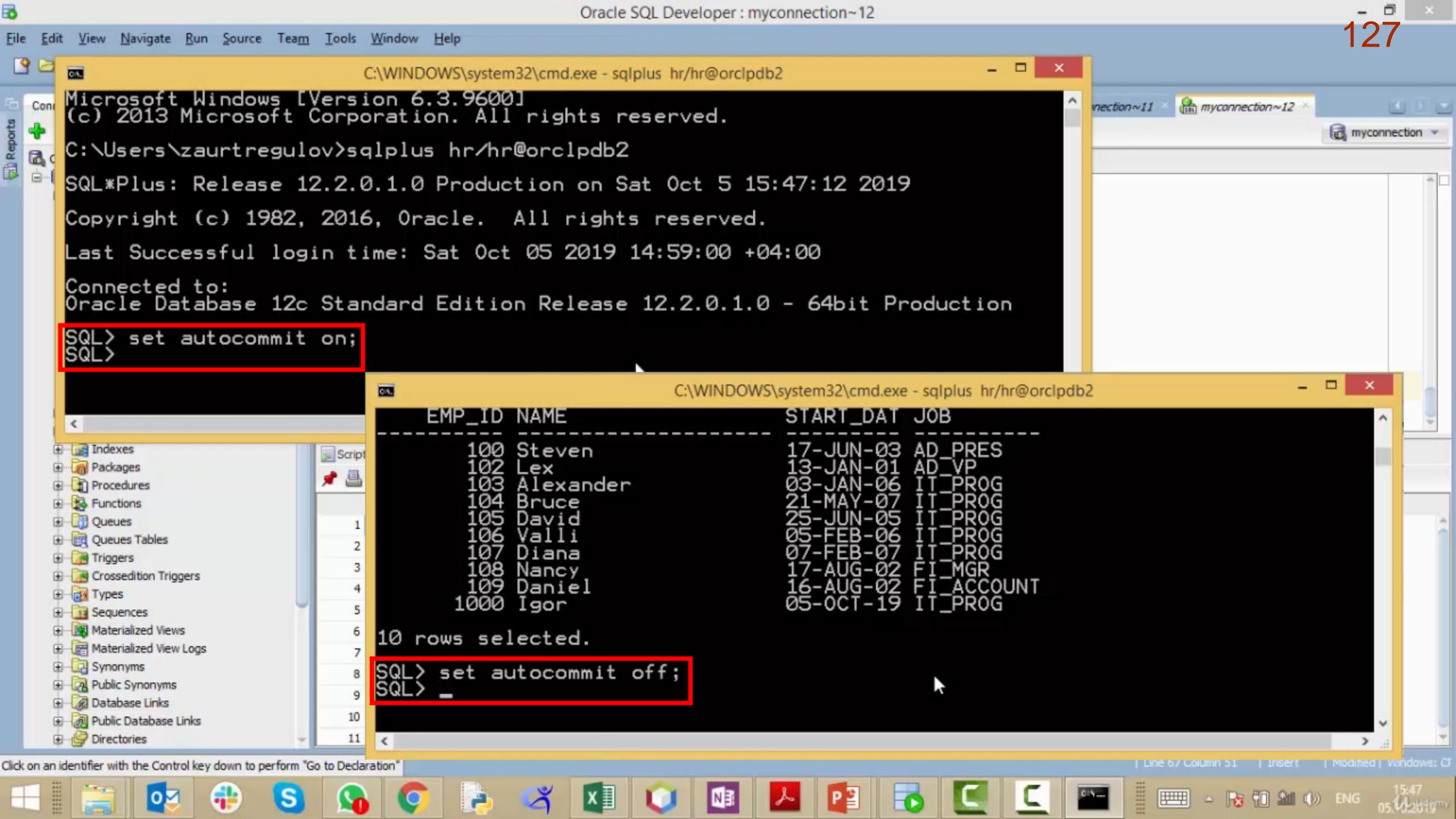


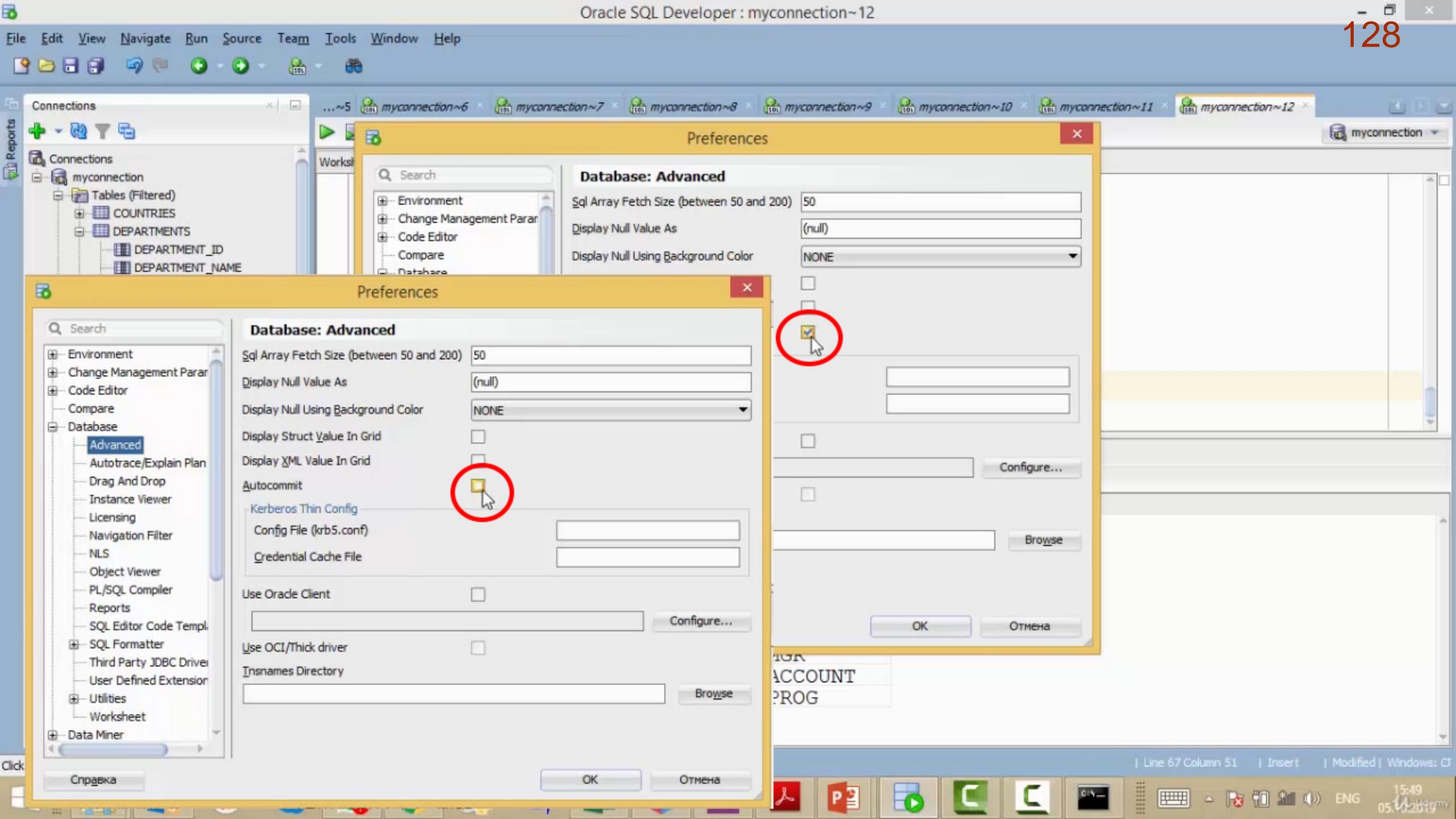
SAVEPOINT

ROLLBACK TO SAVEPOINT savepoint_name;



SAVEPOINT savepoint_name;





Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
DELETE from new_emps where emp_id = 101;  
  
UPDATE new_emps SET emp_id = 300 where emp_id=100;  
  
select * from new_emps;  
UPDATE new_emps SET emp_id = 1000 where emp_id=103;  
  
select * from new_emps for update;
```

Script Output | Query Result | SQL | All Rows Fetched: 8 in 0,003 seconds

EMP_ID	NAME	START_DATE	JOB
1	300 Steven	17-JUN-03	AD PRES
2	104 Bruce	21-MAY-07	IT PROG
3	105 David	25-JUN-05	IT PROG
4	106 Valli	05-FEB-06	IT PROG
5	107 Diana	07-FEB-07	IT PROG
6	108 Nancy	17-AUG-02	FI MGR
7	109 Daniel	16-AUG-02	FI ACCOUNT
8	1000 Iqor	05-OCT-19	IT PROG

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 72 Column 34 | Insert | Modified | Windows: C



```
9 rows selected.

SQL> delete from new_emps where emp_id=103;
1 row deleted.

SQL> commit;

Commit complete.

SQL> delete from new_emps where emp_id=105;
1 row deleted.

SQL> commit;

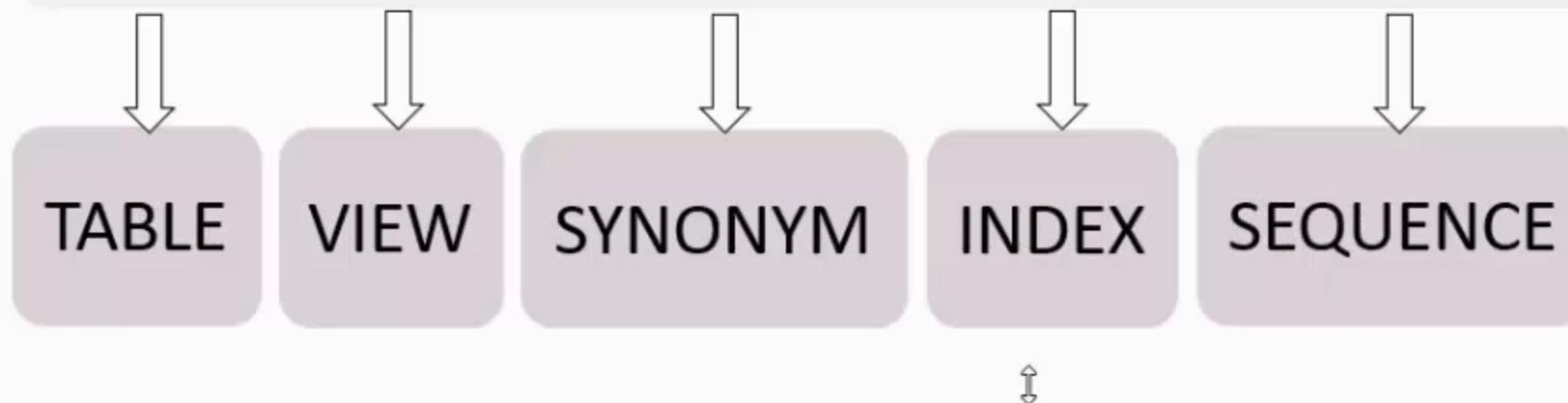
Commit complete.

SQL> select * from new_emps where emp_id > 9
```

```
SQL> select * from new_emps where emp_id > 900 for update;
```

	EMP_ID	NAME	START_DATE	JOB
1	300	Steven	17-JUN-03	AD PRES
2	1500	Bruce	21-MAY-07	IT PROG
3	106	Valli	05-FEB-06	IT PROG
4	107	Diana	07-FEB-07	IT PROG
5	108	Nancy	17-AUG-02	FI MGR
6	109	Daniel	16-AUG-02	FI ACCOUNT
7	1000	Igor	05-OCT-19	IT PROG

DB OBJECTS



USERS SCHEMA

132

USERS SCHEMA
username
pwd

Записная книжка... > Заметки на полях

DML
TRANSA
UPDATE

Trans
ACID

emp
Update

...

(dept->d) Sum(salary)

username

pwd

table 1

Index 13

2ans

table 2

timer

PES

0 0 0 0



USERS SCNF
имя
PWD
Записная книжка... > Заметки на полях

DML TRANS

UPDATE

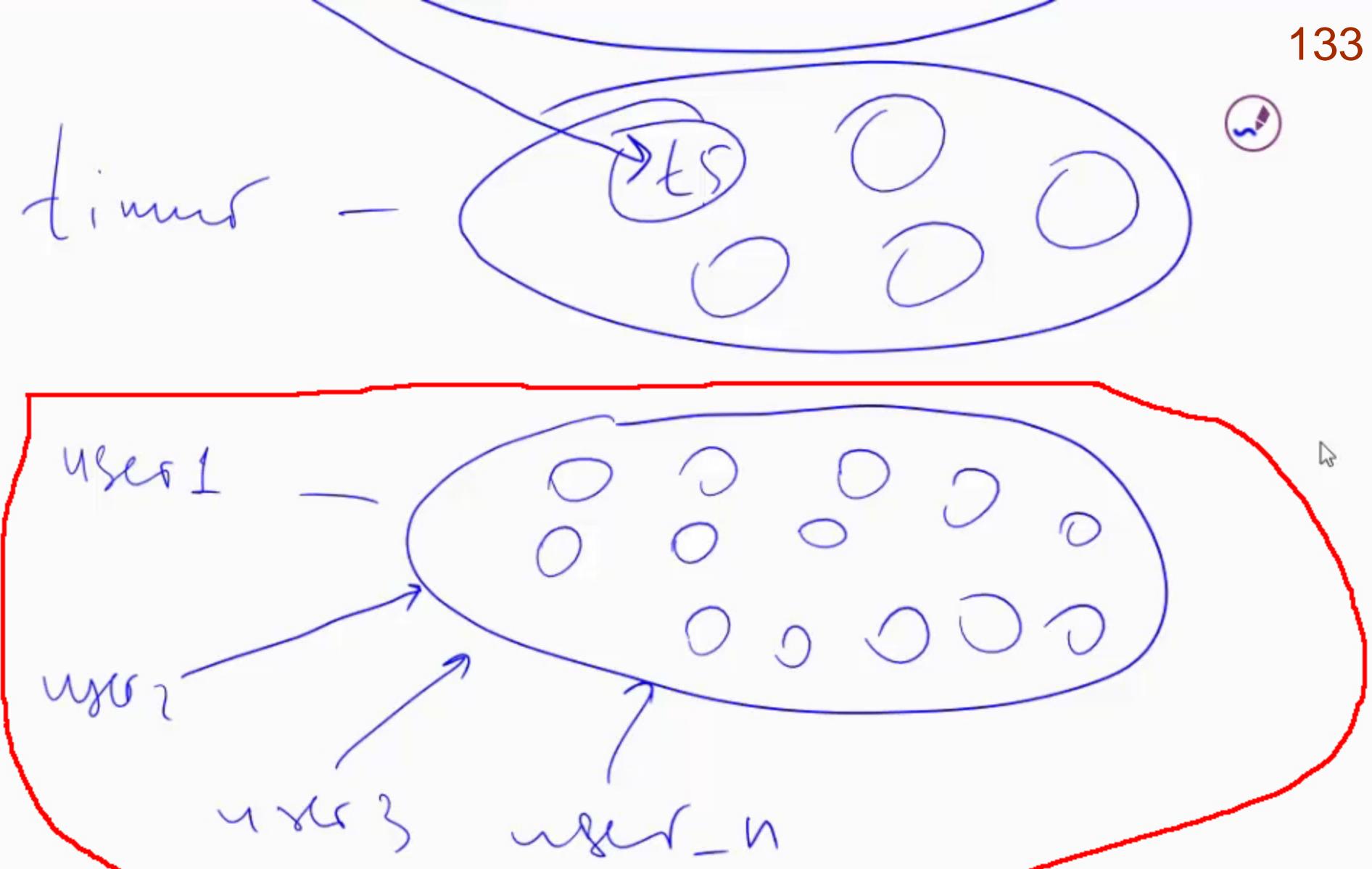
Trans

TRANSA
(ACID)

emp

Update

dept_id Sum(salary)



USERS SCHE
username
PWD

Записная книжка... > Заметки на полях

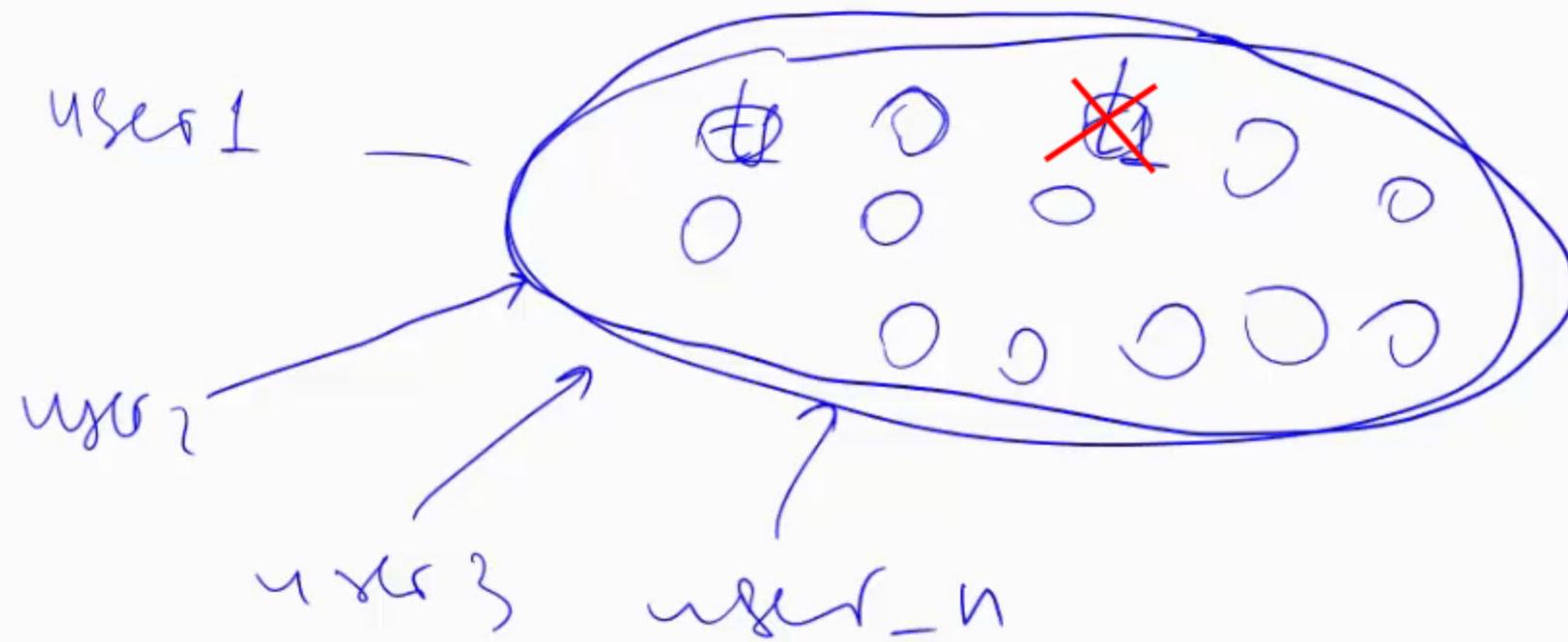
DML
TRANSA
UPDATE

Trans
TRANSA
ACID

emp
Update
1

...

(dept->d) Sum(salary)



schems-name . object-name

USERS SCHE
username
password

Записная книжка... > Заметки на полях

DML
TRANSA
UPDATE

Trans
TRANSA
ACID

emp
Update
07

...

(dept_id) Sum(salary)

user3 user_n
users.t1
schemas-name object-name



SYS
SYSTEM
admin
moni for my
data dictionary
~~DM~~ DDL



ПРАВИЛА НАЗВАНИЯ ОБЪЕКТОВ

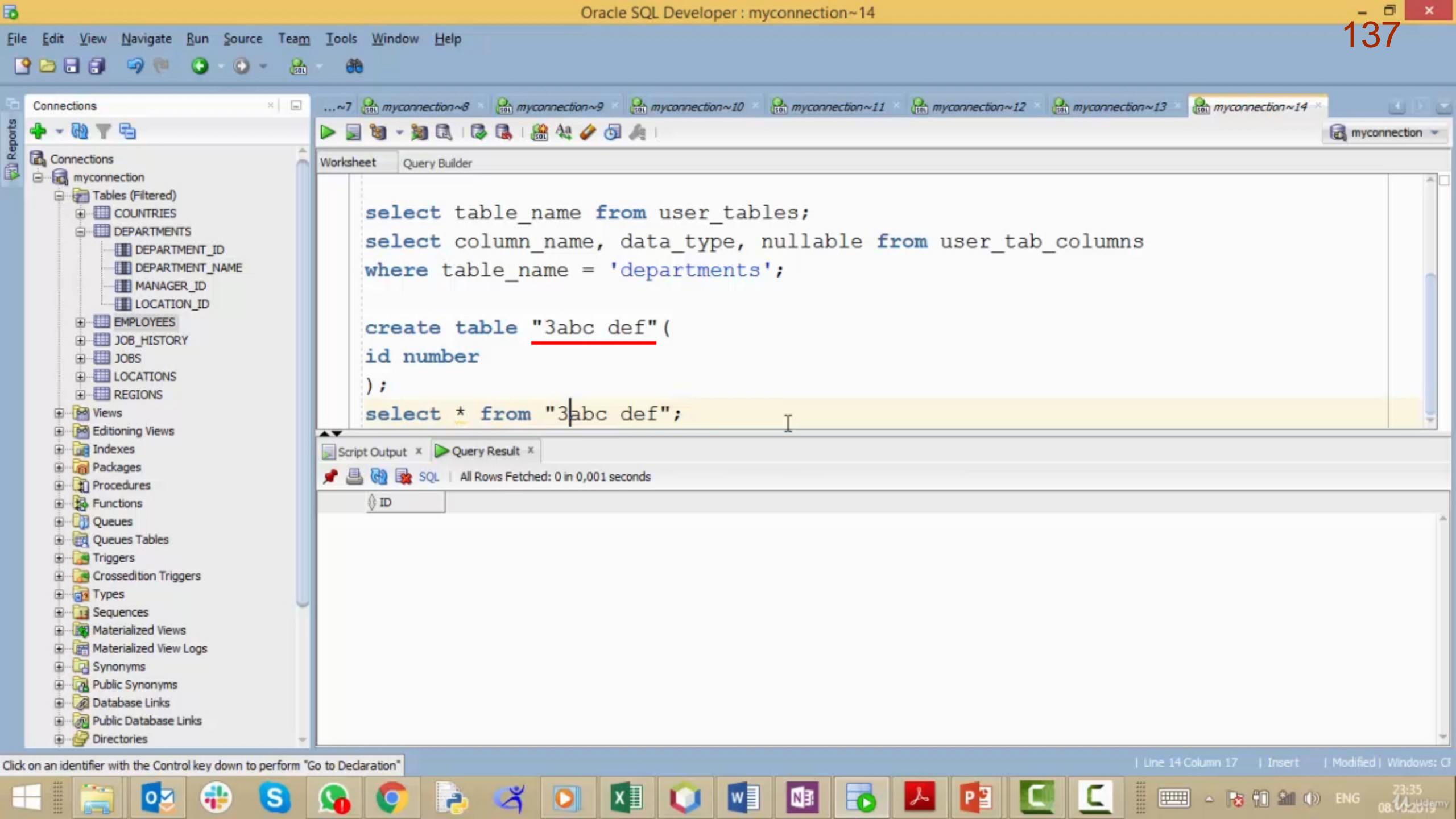
Длина названия должна быть от 1 до 30 символов

Нельзя использовать зарезервированные слова

Названия должны начинаться с букв

Помимо букв в названии можно использовать цифры и символы «_», «#», «\$»

Прописные буквы будут автоматически конвертированы в заглавные



NAMESPACE

TABLE

SEQUENCE

PRIVATE
SYNONYM

VIEW

INDEX

CONSTRAINT

PUBLIC
SYNONYM

ДРУГИЕ ТИПЫ ДАННЫХ

TIMESTAMP WITH TIMEZONE

TIMESTAMP WITH LOCAL TIMEZONE

INTERVAL YEAR TO MONTH

INTERVAL DAY TO SECOND

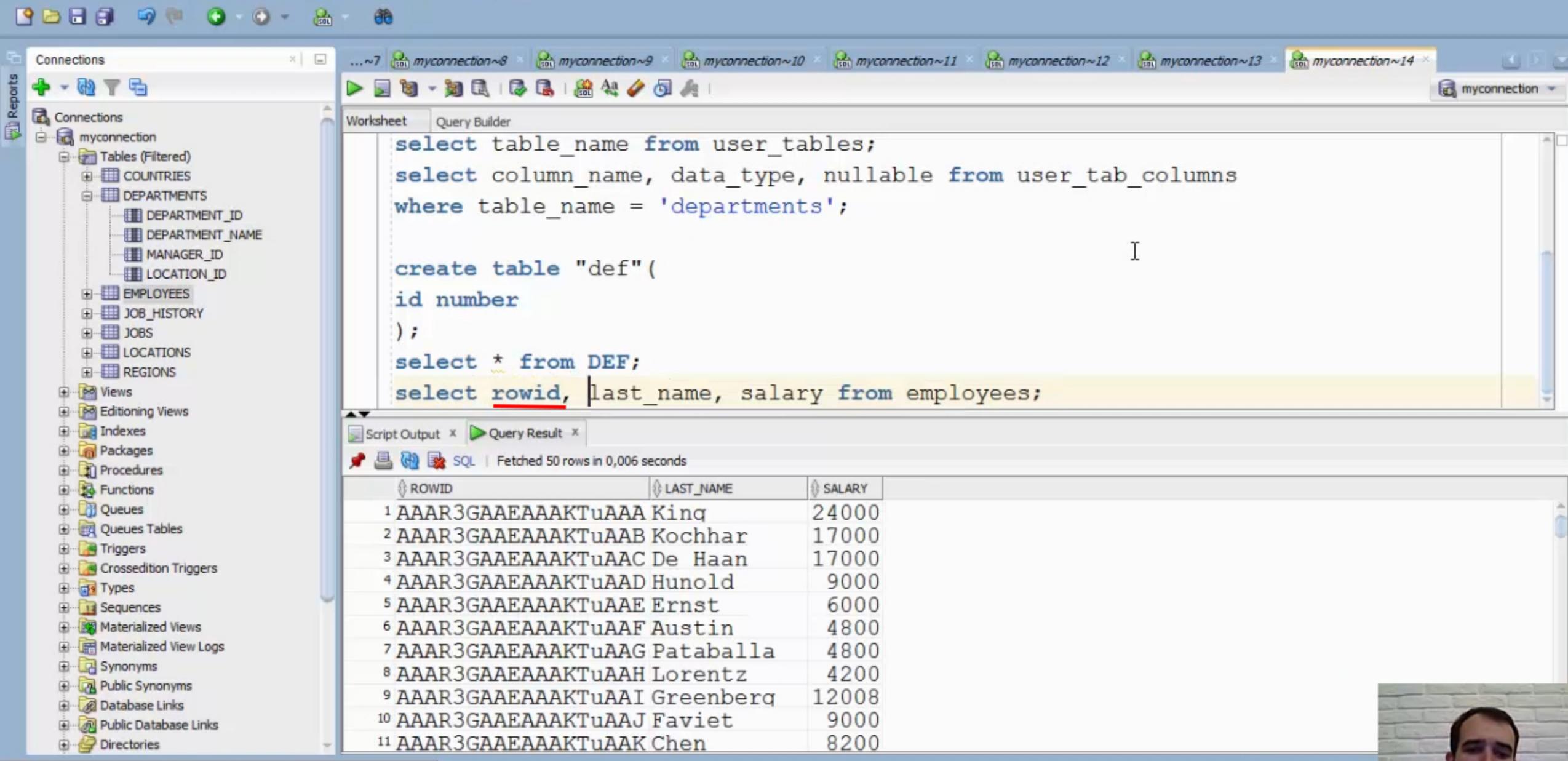
CLOB

BLOB

LONG

ROWID





Click on an identifier with the Control key down to perform "Go to Declaration".



TABLE CREATION

I

```
CREATE TABLE schema.table ORGANIZATION HEAP  
  (column_name datatype DEFAULT expr,  
   column_name datatype DEFAULT expr,  
    ...);
```

TABLE CREATION

```
CREATE TABLE schema.table ORGANIZATION HEAP  
  (column_name datatype DEFAULT expr,  
   column_name datatype DEFAULT expr,  
    ...);
```

```
CREATE TABLE schema.table  
  AS  
  subquery;
```

ALTERING TABLE

```
ALTER TABLE table_name  
ADD (column_name data_type DEFAULT expr);
```

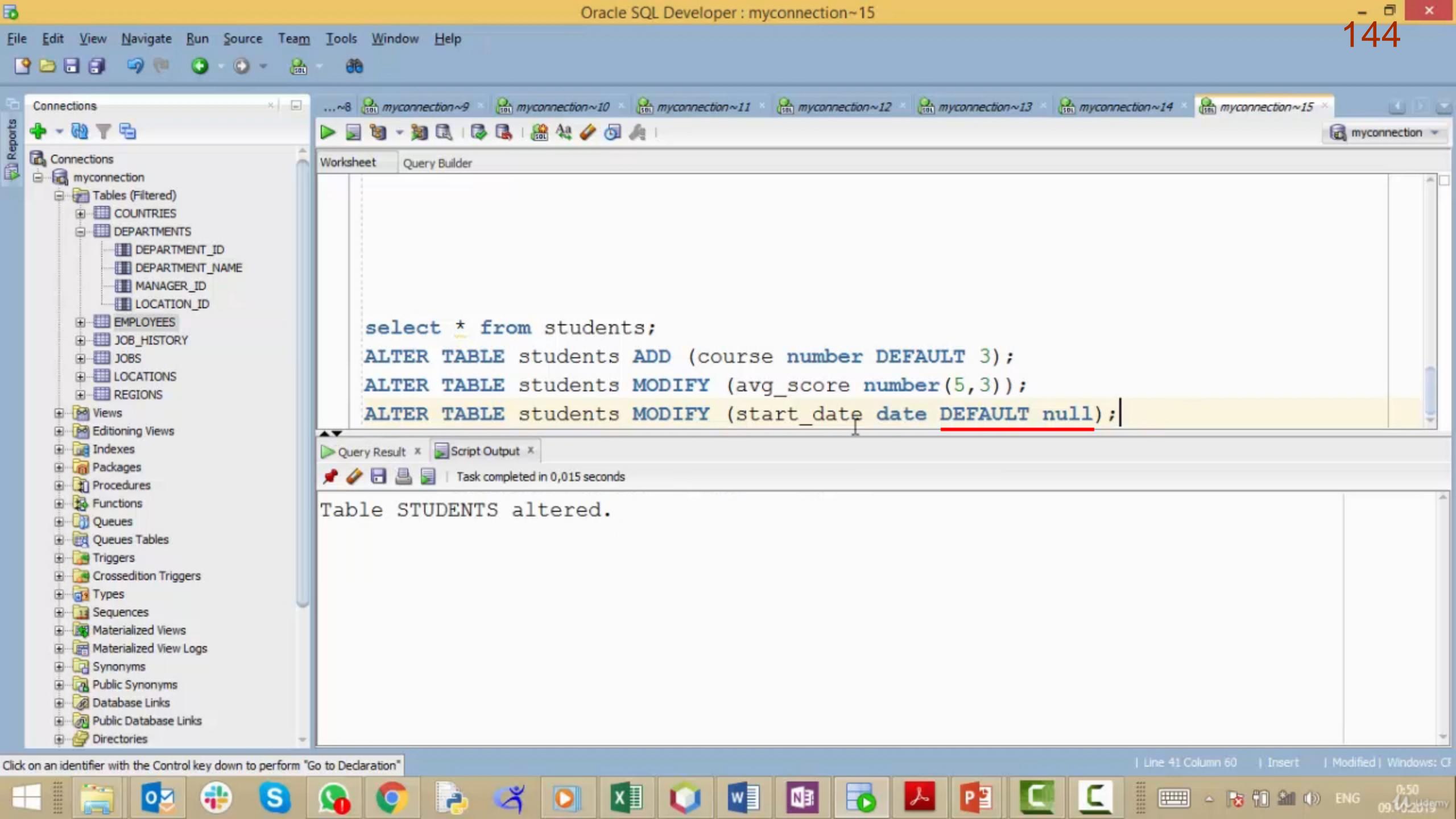
```
ALTER TABLE table_name  
MODIFY (column_name data_type DEFAULT expr);
```

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

```
ALTER TABLE table_name  
SET UNUSED COLUMN column_name;  
ALTER TABLE table_name DROP UNUSED COLUMNS;
```

```
ALTER TABLE table_name  
RENAME COLUMN column_name1 TO column_name2;
```

```
ALTER TABLE table_name READ ONLY;
```



TRUNCATE

```
TRUNCATE TABLE  
schema.table_name;
```



☰ Последние

+ Страница

extent 1 

Записная книжка... > Заметки на полях

1) 1-30 символов
new-caps -
 $a - b - \dots$

USERS SCHE
username
pwd

DML TRANS
UPDATE 11

Trans
ACID

emp Update

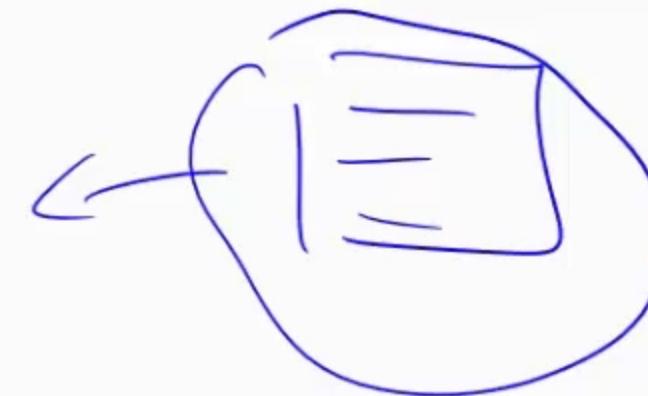
9 октября 2019 г. 1:20

extent 1

extent 2

- - -

extent N



high water mark

ex 1



ex 2



ex 3



☰ Последние

+ Страница

extent 1 ← (≡)

Записная книжка... > Заметки на полях

1) 1-30 символов
new-caps -
→ a - b - z -

USERS SCHE
имя
PWD

DML TRANS
UPDATE 11

Trans
ACID

emp Update

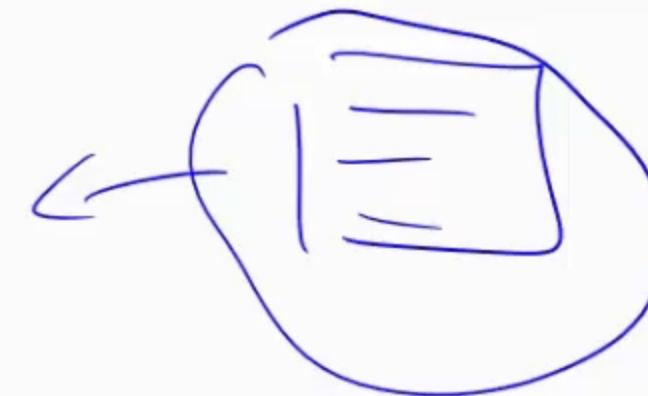
9 октября 2019 г. 1:20

extent 1

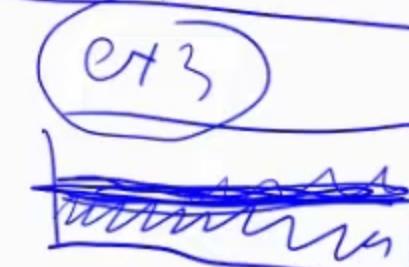
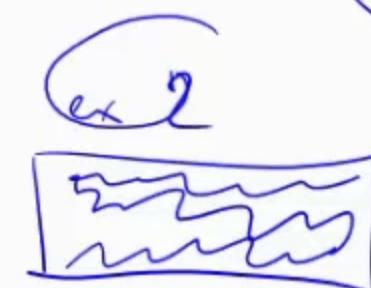
extent 2

- - -

extent N



high water mark



TRUNCATE

DROP

```
DROP TABLE  
schema.table_name;
```



Students
 id name course faculty avg-score start-date
 Записная книжка... > Заметки на полях

extent 1
 extent 2

1) 1-30 символов
 new-cards -
 \ a - b - 1 -

USERS SCRF
 login
 pwd

DML
 TRANS
 UPDATE

TRANSAC
 ACID
 Finsat



~~Faculty~~

id	name
1	CS
2	Economics
3	Philology

- 1) not null
- 2) primary key }
 3) foreign key }
 4) check }
 5) unique }

I

primary key
unique

[INDEX]

Записная книжка... > Заметки на полях

~~students~~

~~id name course family~~

~~extent1~~ ↗
~~extent2~~

1) 1-30 combinations
new-cards -
a - b - 1

~~USERS SCRF~~

~~username~~
~~pwd~~

DML
TRANSA

UPDATE t1

primary key
unique

[INDEX]

id	name
- 1 -	- - -
- - -	- - -
- - -	- - -
100000	- - -



INDEX

select * from j,
where id = 523846;
"where" - 452 op.
1-957



B-tree
+ balanced

primeng
unique

[EDIT]

Записная книжка... > Заметки на полях

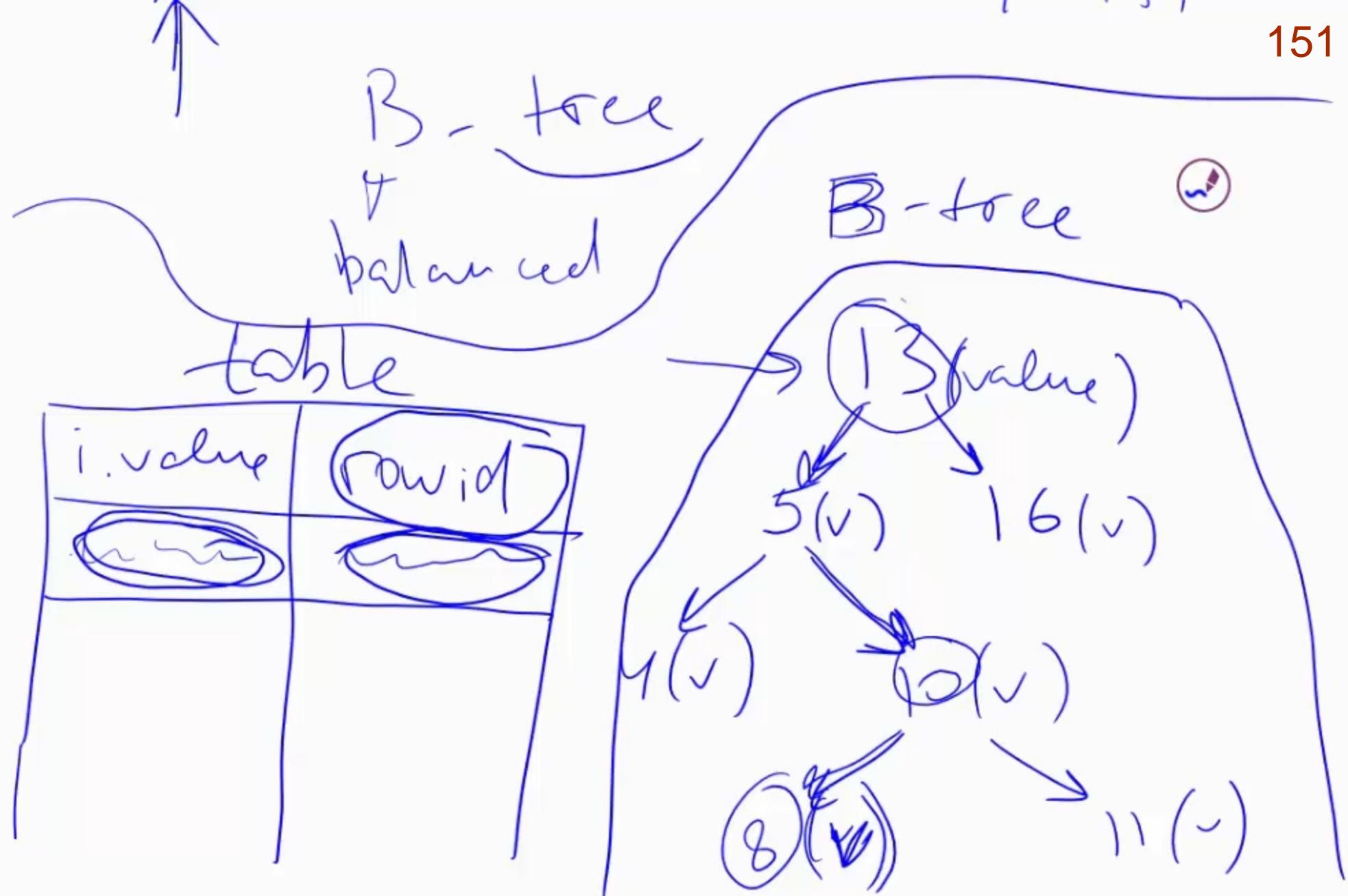
~~students~~
 id name course family_group

extent1 extent2

1) 1-30 combinations
 new-cards -
 ↗ a - b - 1

USERS SCRF
 username
 pwd

DML
 TRANS
 UPDATE t1



+ Страница

регистрация
уникальный

INT

Записная книжка... > Заметки на полях

students

id name course family_name

extent1 extent2

1) 1 - 30 символов
new-cards -
\ a - b - 1

USERS SCRF

username
pwd

DML TRANS

UPDATE t1



CONSTRAINT TYPES

UNIQUE принуждает столбец(цы) содержать только уникальные значения. Исключение – null.



Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number CONSTRAINT st_id_unique UNIQUE,
    name varchar2(15),
    course number,
    faculty_id integer,
    avg_score number(5, 2),
    start_date date,
    scholarship integer
);
```

Script Output X

Task completed in 0,066 seconds

)

Error report -

SQL Error: ORA-00922: missing or invalid option
00922. 00000 - "missing or invalid option"

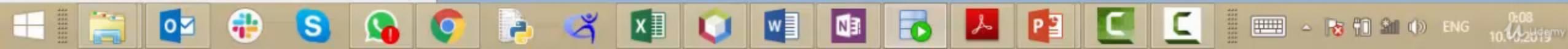
*Cause:

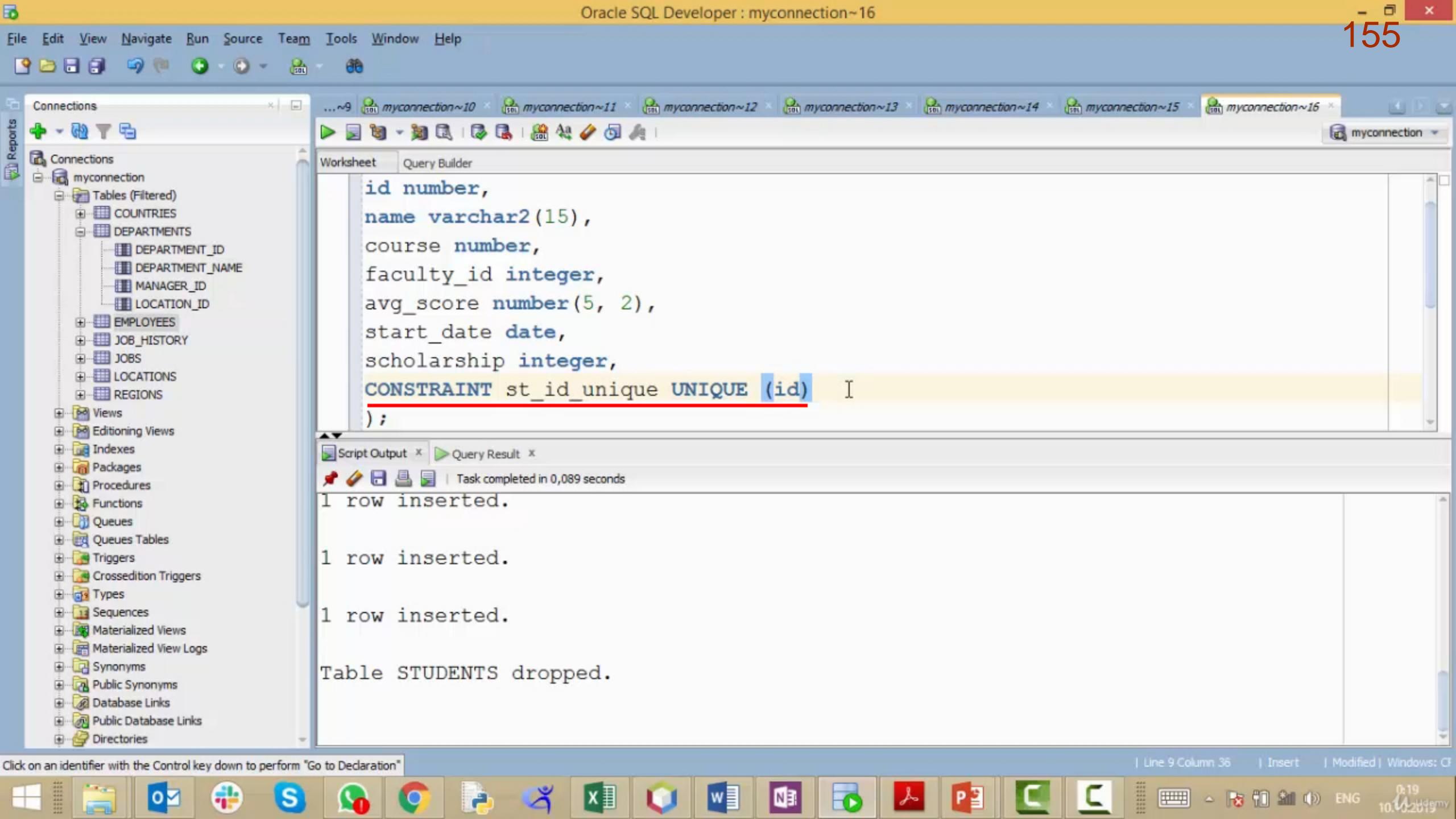
*Action:

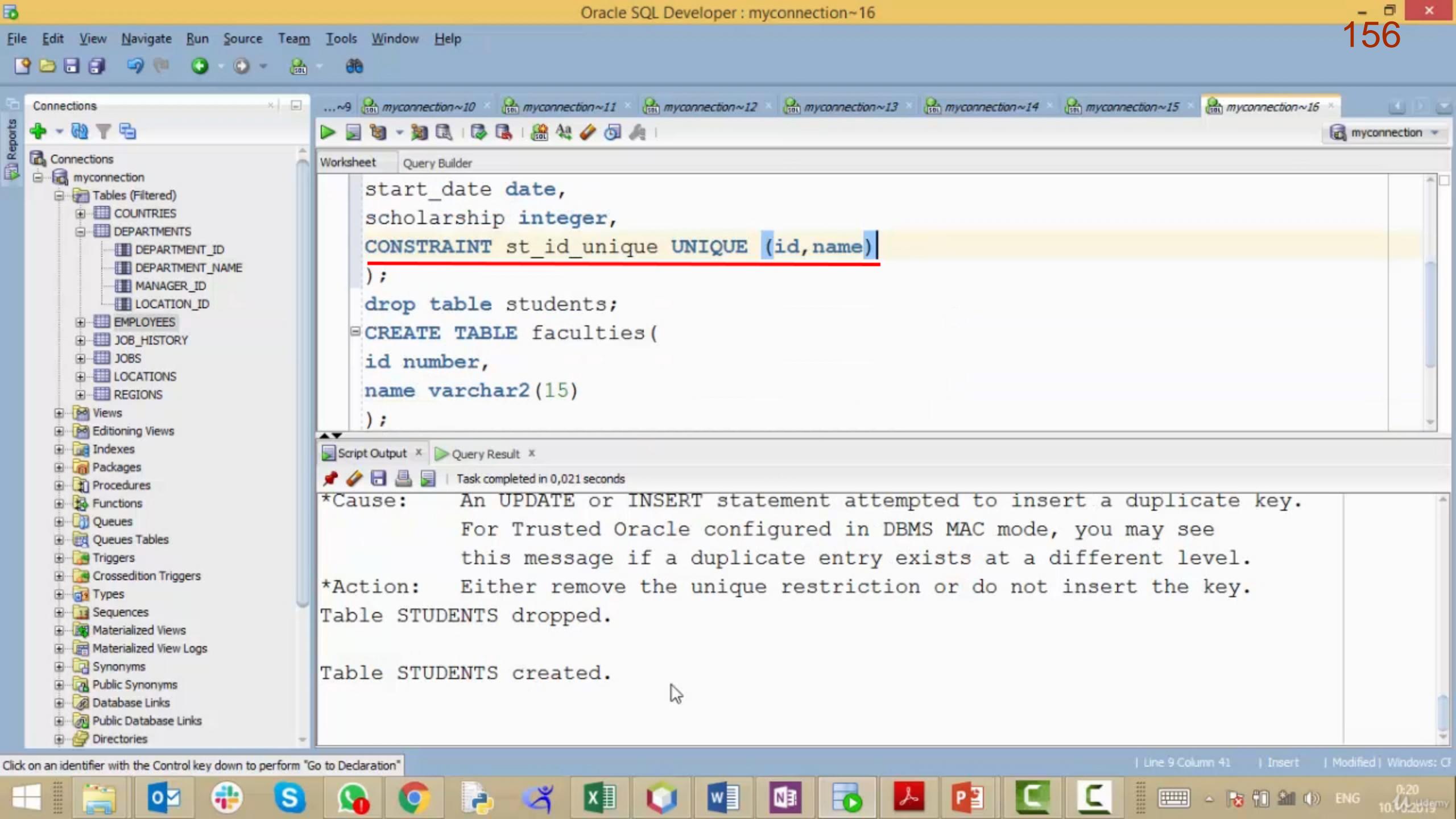
Table STUDENTS created.

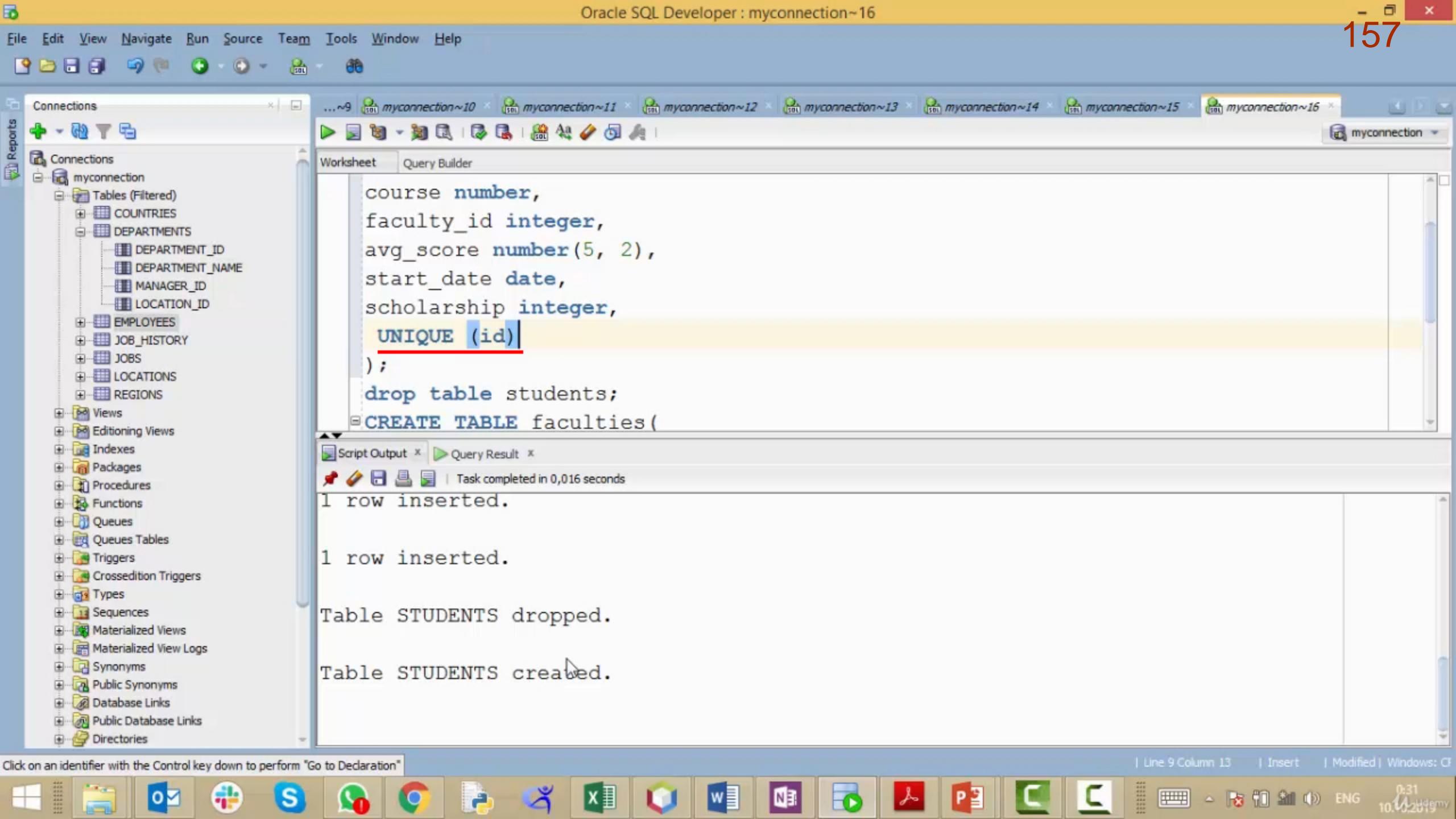
Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 9 Column 3 | Insert | Modified | Windows: C









File Edit View Navigate Run Source Team Tools Window Help

Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
id number UNIQUE,  
name varchar2(15),  
course number,  
faculty_id integer,  
avg_score number(5, 2),  
start_date date,  
scholarship integer  
);  
drop table students;
```

Script Output x Query Result x

Task completed in 0,014 seconds

)

Error report -

SQL Error: ORA-00904: : invalid identifier
00904. 00000 - "%s: invalid identifier"

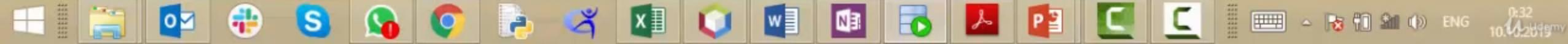
*Cause:

*Action:

Table STUDENTS created.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 6 Column 24 | Insert | Modified | Windows: C



Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```

name varchar2(15)
);

select * from students;

INSERT INTO students values (1, 'Zaur', 3, 1, 8.7, TO_DATE('15-SEP-17'), 1500);
INSERT INTO students values (1, 'Kolya', 3, 1, 8.7, TO_DATE('15-SEP-17'), 1500);
INSERT INTO students values (2, 'Misha', 2, 3, 7.5, TO_DATE('15-SEP-18'), 800);
INSERT INTO students values (null, 'Zaur', 3, 1, 8.7, TO_DATE('15-SEP-17'), 1500);

alter table students| add constraint st_id_unique UNIQUE(id);

```

Script Output x | Task completed in 0,032 seconds

Query Result x

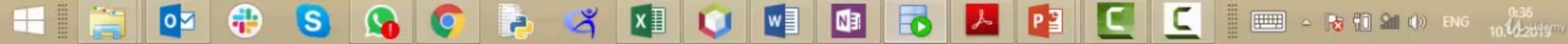
```

alter table student add constraint st_id_unique UNIQUE(id)
Error report -
SQL Error: ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"
*Cause:
*Action:
Table STUDENTS altered.

```

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 21 Column 21 | Insert | Modified | Windows: C



Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```

INSERT INTO students values (null, 'Zaur', 3, 1, 8.7, TO_DATE('15-SEP-17'), 1500);

alter table students add constraint st_id_unique UNIQUE(id);
INSERT into faculties VALUES(1, 'CS');
INSERT into faculties VALUES(2, 'Economics');
INSERT into faculties VALUES(2, 'Philology');
select * from faculties;
alter table faculties add UNIQUE(id);
update faculties set id=3 where name = 'Philology';
  
```

Script Output x Query Result x

Task completed in 0,021 seconds

02299. 00000 - "cannot validate (%s.%s) - duplicate keys found"

*Cause: an alter table validating constraint failed because the table has
duplicate key values.

*Action: Obvious

1 row updated.

Table FACULTIES altered.

Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
alter table students add constraint st_id_unique UNIQUE(id);
INSERT into faculties VALUES(1, 'CS');
INSERT into faculties VALUES(2, 'Economics');
INSERT into faculties VALUES(3, 'Philology');
select * from faculties;
alter table faculties add UNIQUE(id);
update faculties set id=3 where name = 'Philology';

alter table students MODIFY (id constraint abc UNIQUE);
```

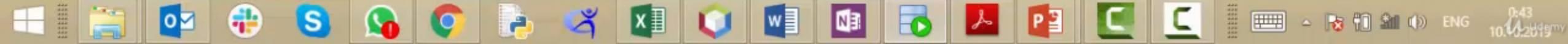
Script Output x Query Result x

Task completed in 0,026 seconds

```
alter table students MODIFY (id constraint abc UNIQUE)
Error report -
SQL Error: ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
Table STUDENTS altered.
```

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 30 Column 17 | Insert | Modified | Windows: C



Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
INSERT into faculties VALUES(1, 'CS');
INSERT into faculties VALUES(2, 'Economics');
INSERT into faculties VALUES(3, 'Philology');
select * from faculties;
alter table faculties add UNIQUE(id);
update faculties set id=3 where name = 'Philology';

alter table students MODIFY (id constraint abc UNIQUE);
alter table faculties MODIFY (id UNIQUE);
```

Script Output x Query Result x

Task completed in 0,02 seconds

SQL Error: ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"

*Cause:

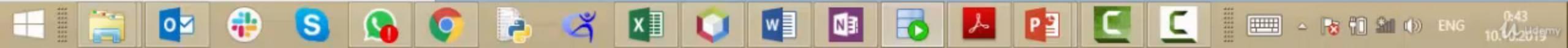
*Action:

Table STUDENTS altered.

Table FACULTIES altered.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 31 Column 43 | Insert | Modified | Windows: C



File Edit View Navigate Run Source Team Tools Window Help

Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
INSERT into faculties VALUES(3, 'Philology');
select * from faculties;
alter table faculties add UNIQUE(id);
update faculties set id=3 where name = 'Philology';

alter table students MODIFY (id constraint abc UNIQUE);
alter table faculties MODIFY (id UNIQUE);

alter table students drop constraint abc;
```

Script Output x Query Result x

ScriptRunner Task

*Cause:

*Action:

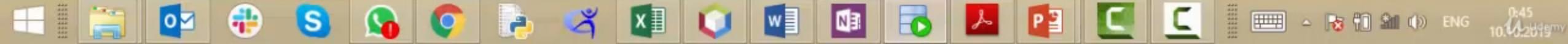
Table STUDENTS altered.

Table FACULTIES altered.

Table STUDENTS altered.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 33 Column 42 | Insert | Modified | Windows: C

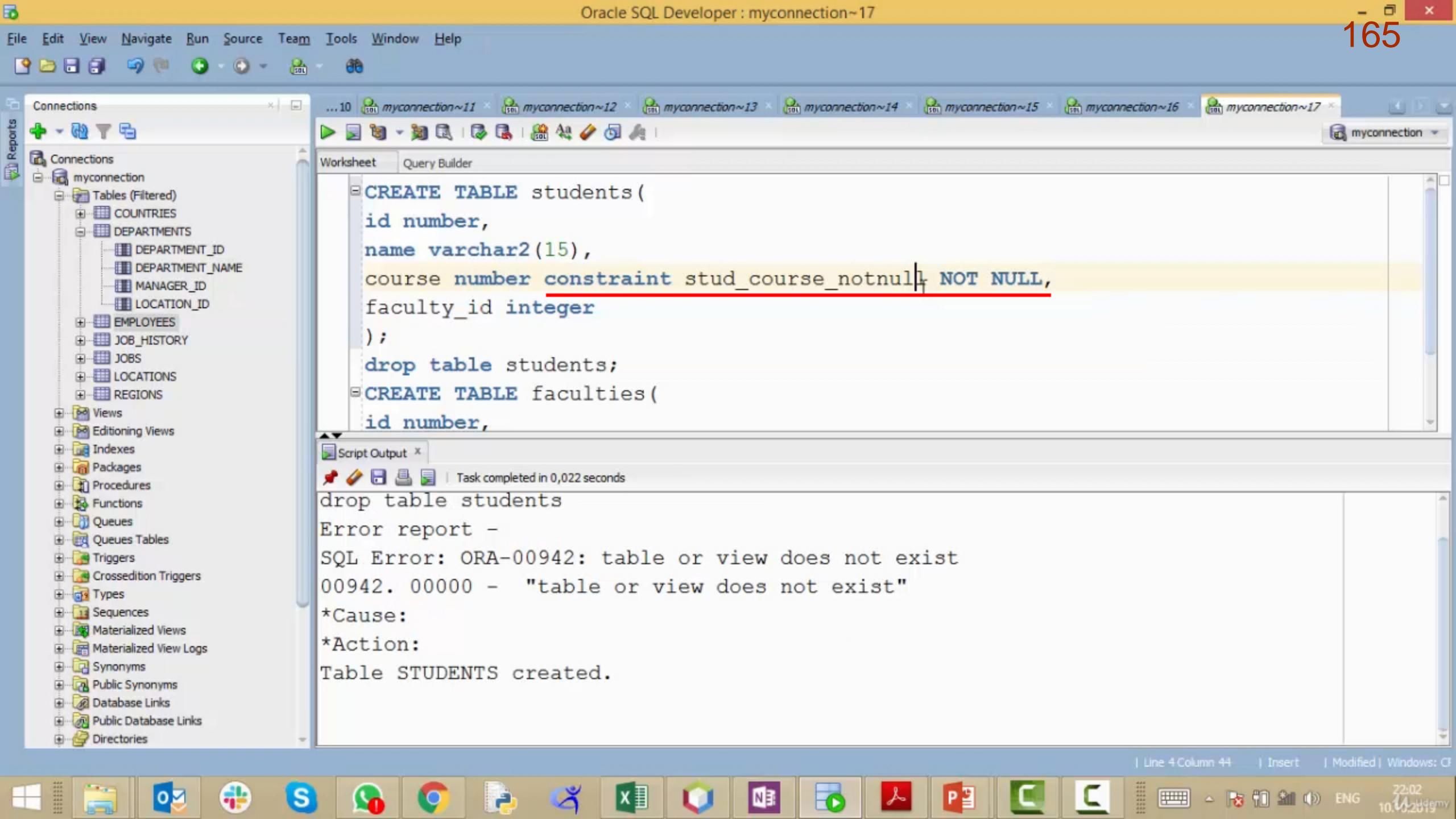


CONSTRAINT TYPES

UNIQUE принуждает столбец(цы) содержать только уникальные значения. Исключение – null.

NOT NULL не разрешает столбцам содержать значение “null”





File Edit View Navigate Run Source Team Tools Window Help



Reports

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number,
    name varchar2(15),
    course number NOT NULL,
    faculty_id integer
);
drop table students;
CREATE TABLE faculties(
    id number,
```

Script Output x Query Result x

Task completed in 0,014 seconds

SQL Error: ORA-01400: cannot insert NULL into ("HR"."STUDENTS"."COURSE")
01400. 00000 - "cannot insert NULL into (%s)"

*Cause: An attempt was made to insert NULL into previously listed objects.

*Action: These objects cannot accept NULL values.

Table STUDENTS dropped.

Table STUDENTS created.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 4 Column 25 | Insert | Modified | Windows: C



File Edit View Navigate Run Source Team Tools Window Help



Reports

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number,
    name varchar2(15),
    course number,
    faculty_id integer,
    constraint abcdef not null (course)
);
drop table students;
CREATE TABLE faculties(
    faculty_id integer,
    constraint abcdef not null (course)
)
```

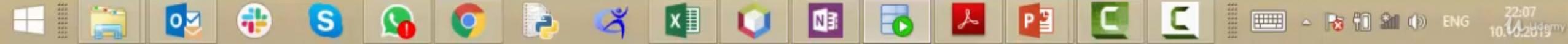
Script Output x Query Result x

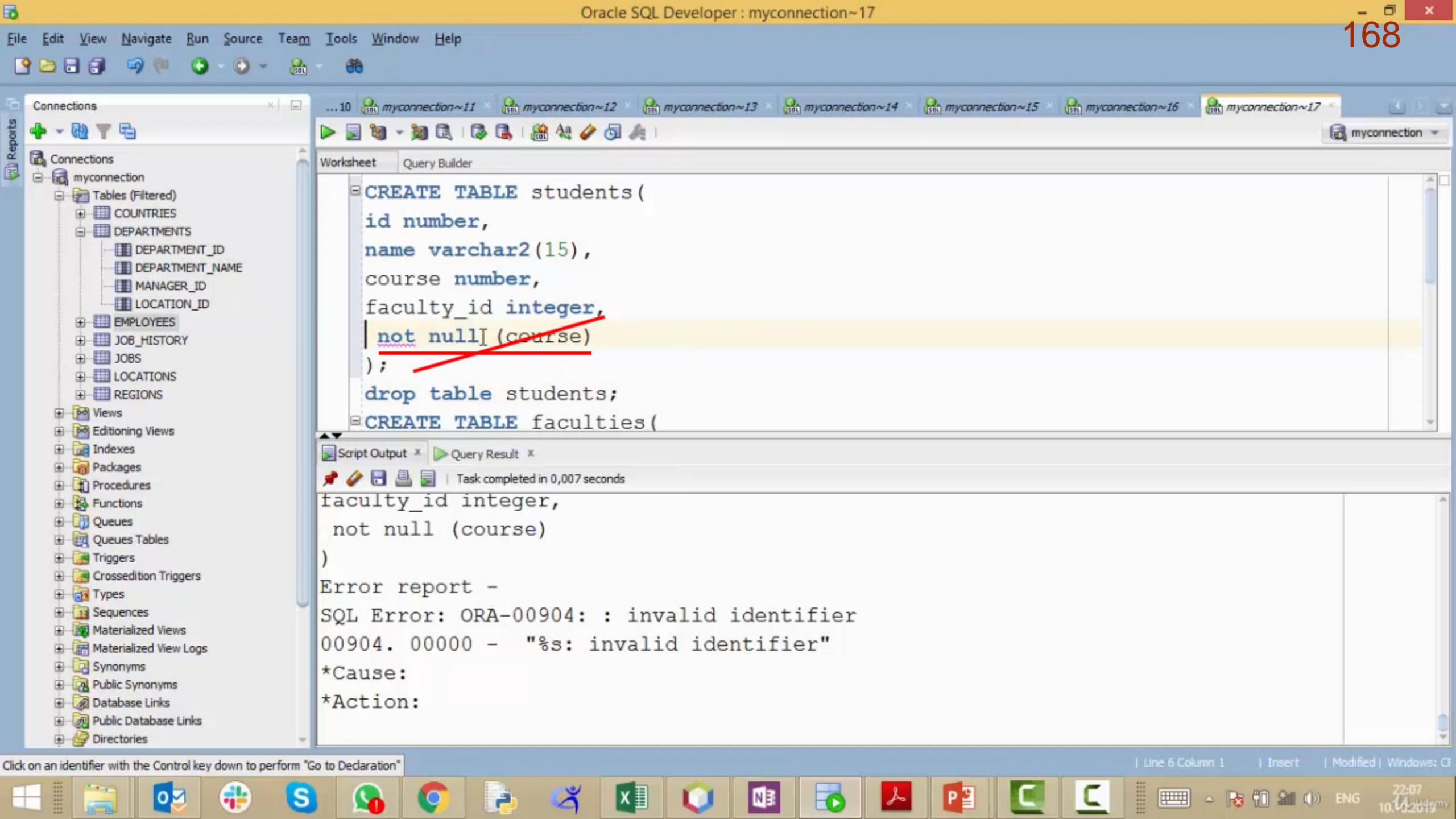
Task completed in 0,007 seconds

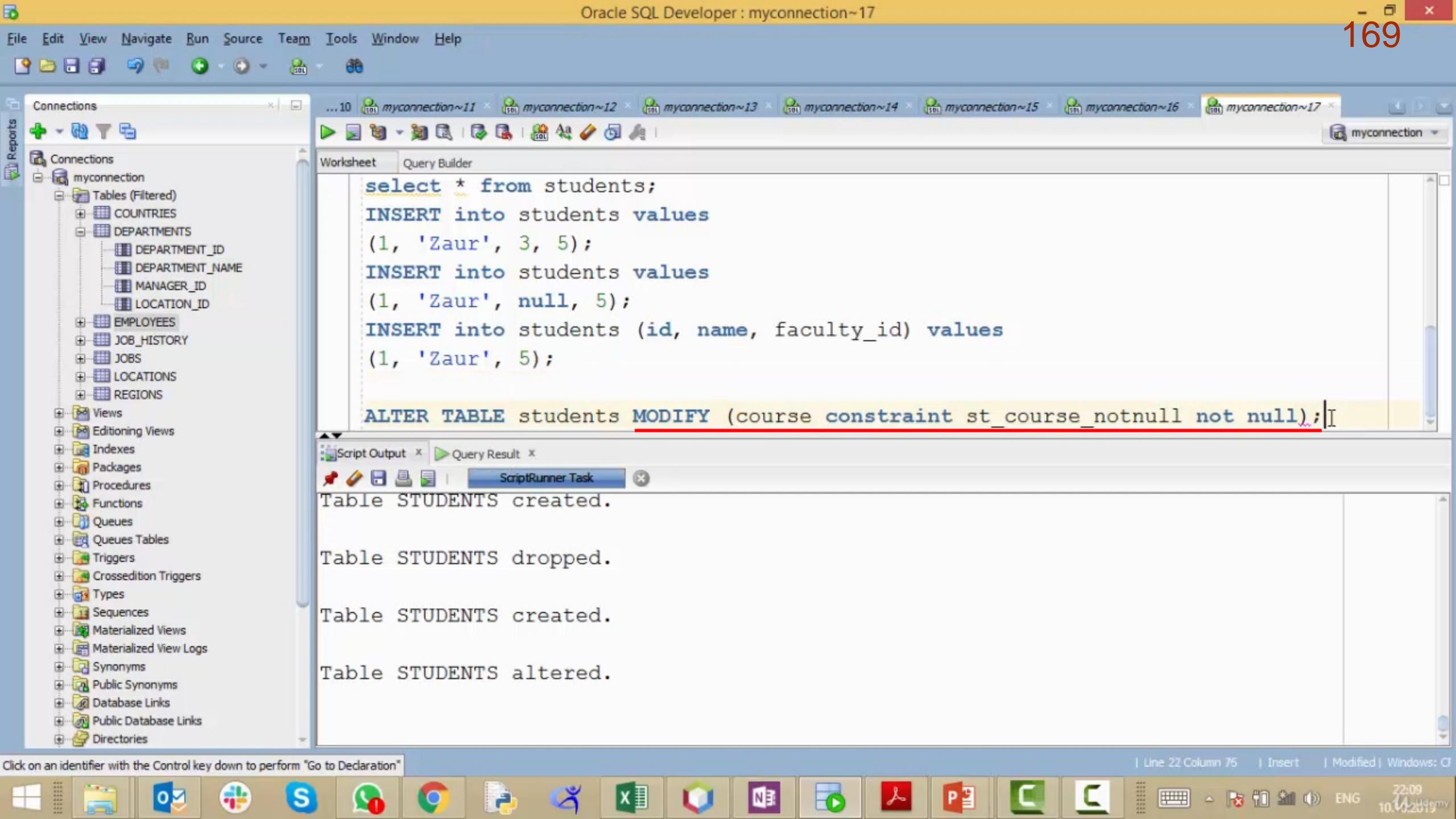
faculty_id integer,
constraint abcdef not null (course)
)
Error report -
SQL Error: ORA-00904: : invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:

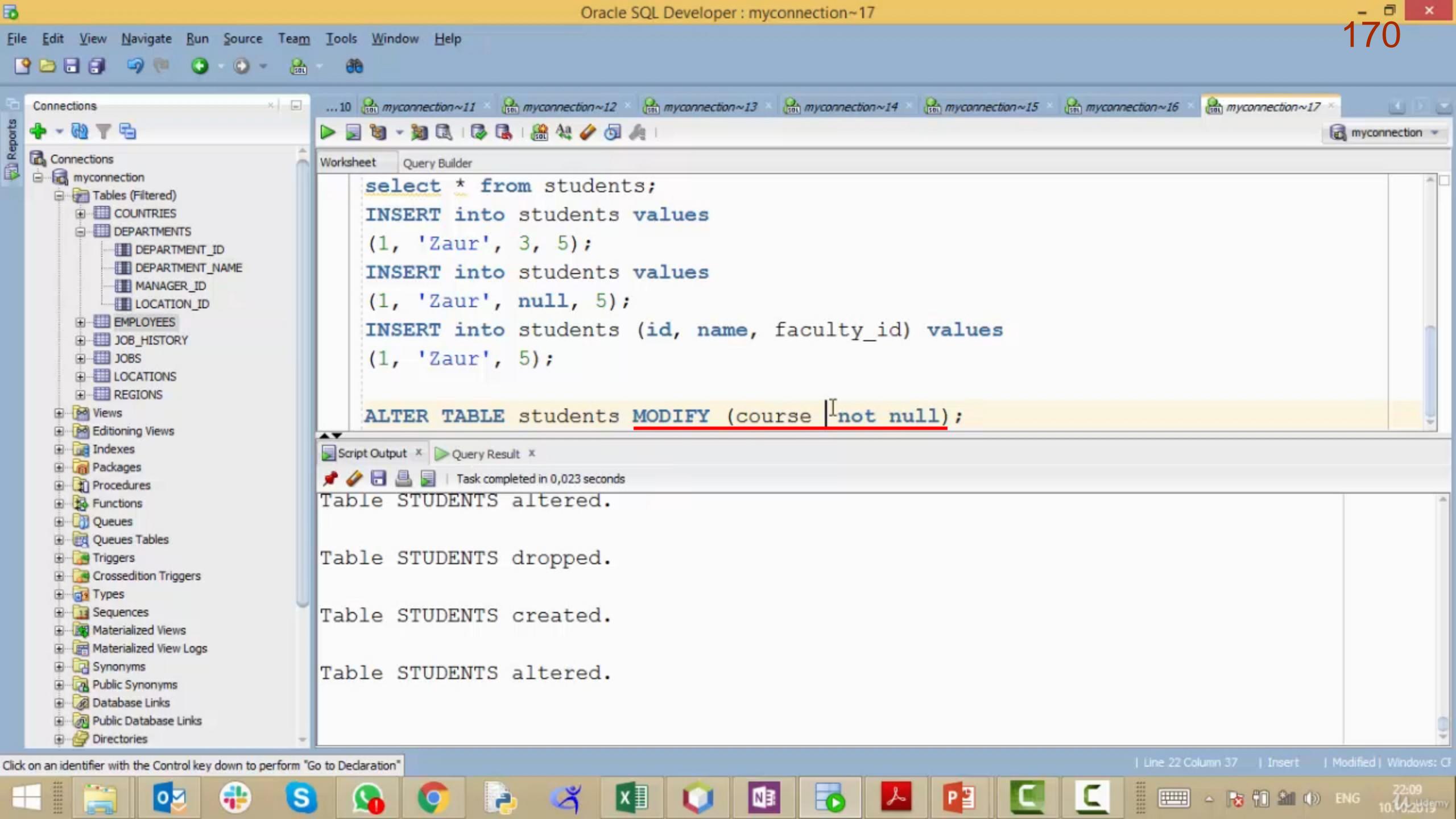
Click on an identifier with the Control key down to perform "Go to Declaration"

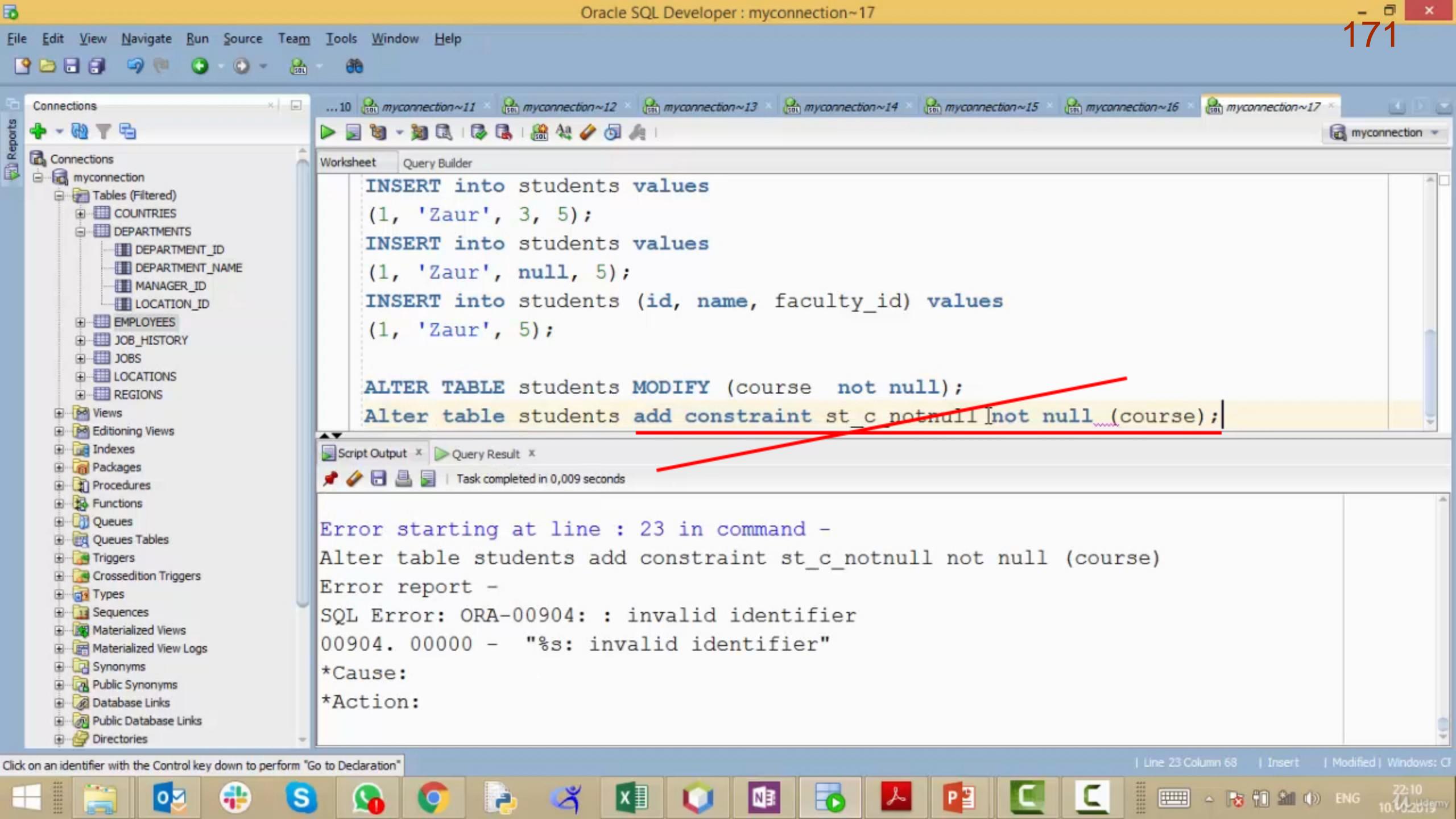
| Line 5 Column 20 | Insert | Modified | Windows: C











CONSTRAINT TYPES

UNIQUE принуждает столбец(цы) содержать только уникальные значения. Исключение – null.

NOT NULL не разрешает столбцам содержать значение “null”

PRIMARY KEY принуждает столбец(цы) содержать только уникальные значения и не разрешает содержать значение “null”



File Edit View Navigate Run Source Team Tools Window Help



Reports

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number constraint st_id_pk primary key,
    name varchar2(15),
    course number ,
    faculty_id integer
);
drop table students;
CREATE TABLE faculties(
    id number,
```

Script Output x Query Result x

Task completed in 0,022 seconds

Table STUDENTS altered.

1 row inserted.

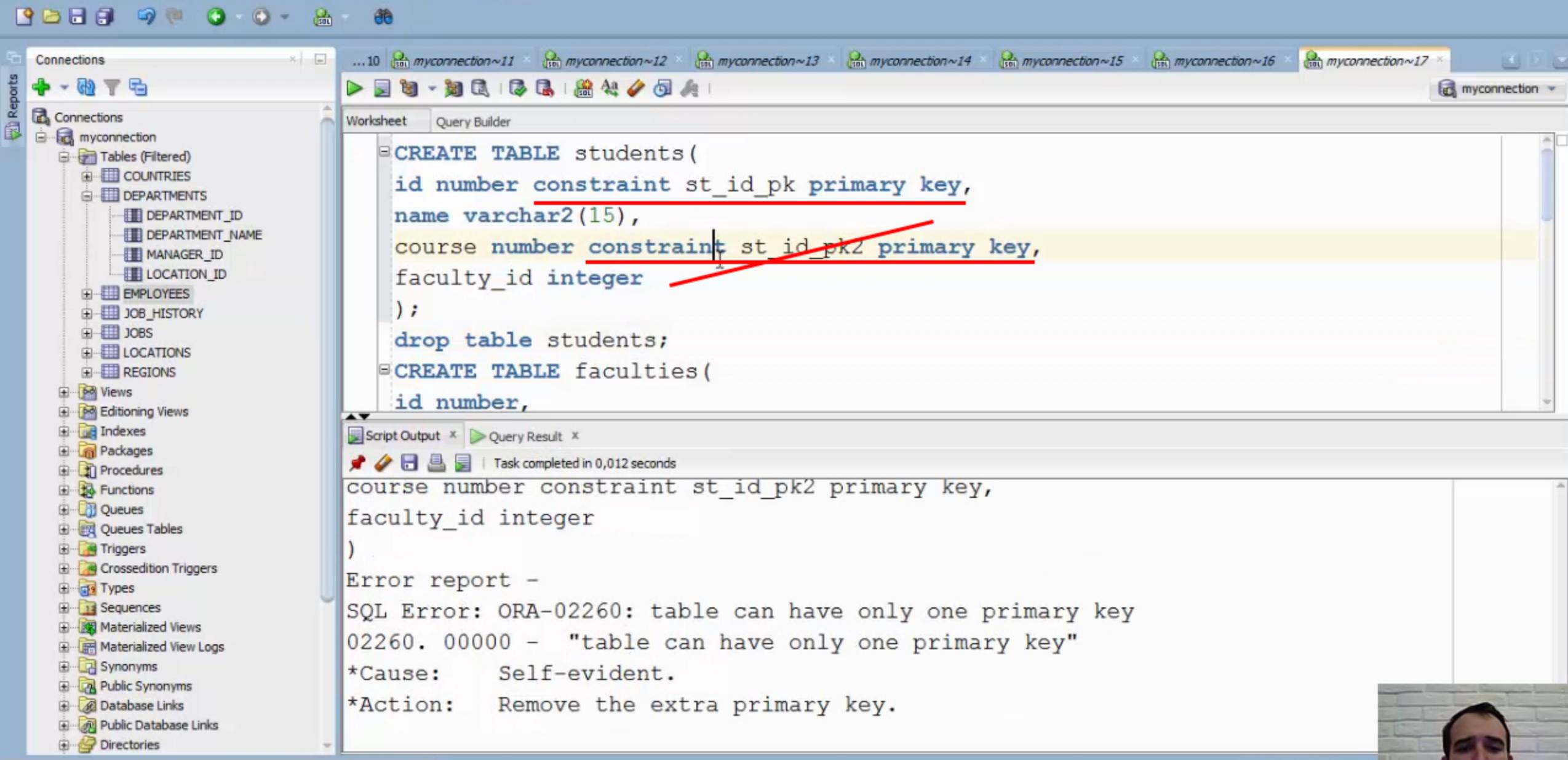
Table STUDENTS dropped.

Table STUDENTS created.

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 2 Column 42 | I





File Edit View Navigate Run Source Team Tools Window Help

Reports

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number primary key,
    name varchar2(15),
    course number ,
    faculty_id integer
);
drop table students;
CREATE TABLE faculties(
    id number,
```

Script Output x Query Result x

Task completed in 0,019 seconds

)

Error report -

SQL Error: ORA-02260: table can have only one primary key
02260. 00000 - "table can have only one primary key"

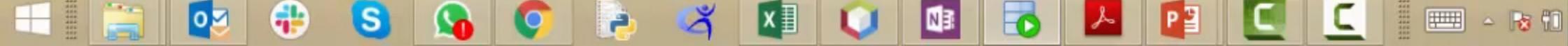
*Cause: Self-evident.

*Action: Remove the extra primary key.

Table STUDENTS created.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 2 Column 10 | I



File Edit View Navigate Run Source Team Tools Window Help



Connections

+ Connections

myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

```
CREATE TABLE students(
    id number ,
    name varchar2(15),
    course number ,
    faculty_id integer,
    constraint pk15 primary key (id)
);
drop table students;
CREATE TABLE faculties(
```

Script Output x Query Result x

Task completed in 0,018 seconds

SQL Error: ORA-00955: name is already used by an existing object

00955. 00000 - "name is already used by an existing object"

*Cause:

*Action:

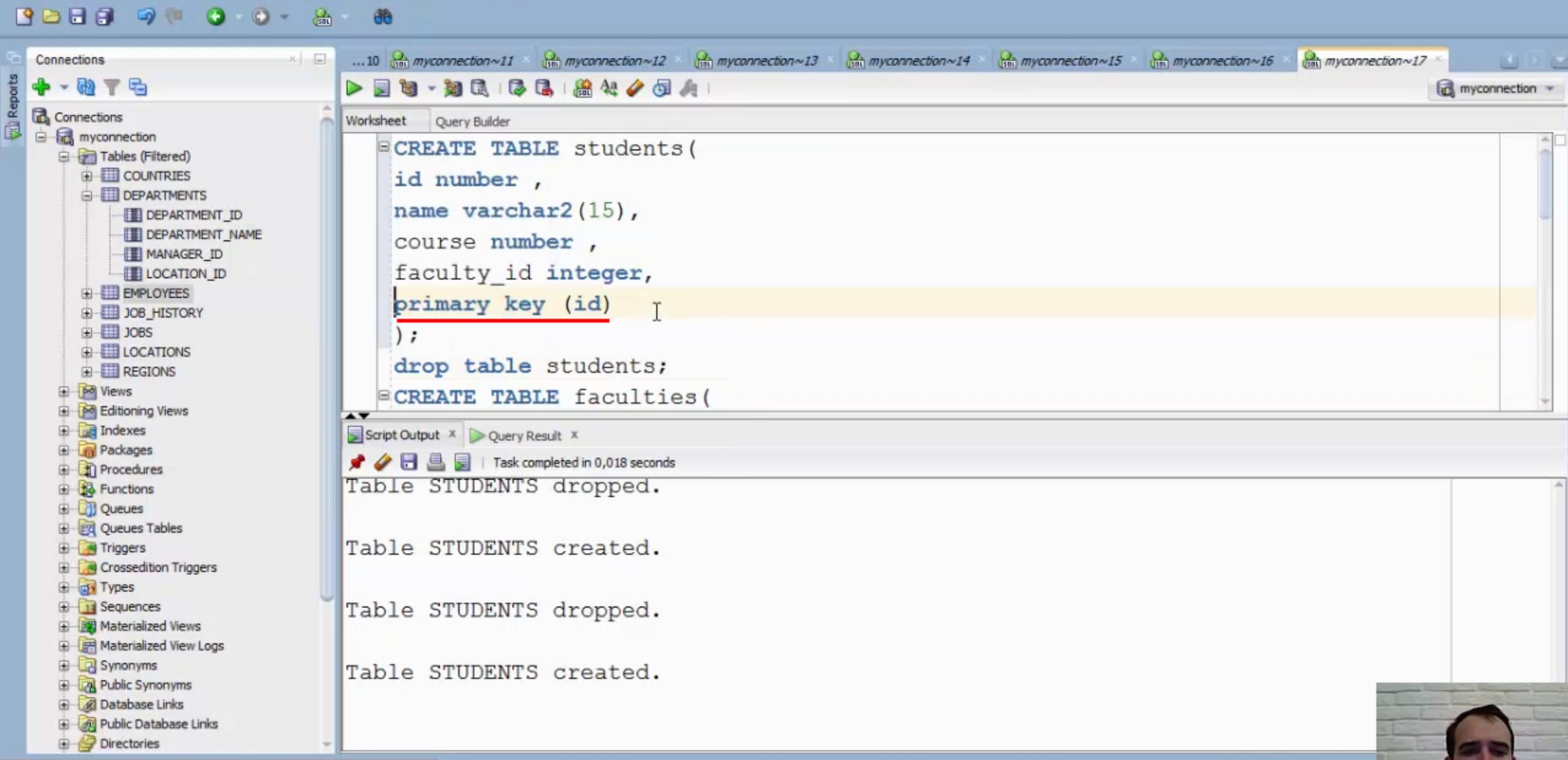
Table STUDENTS dropped.

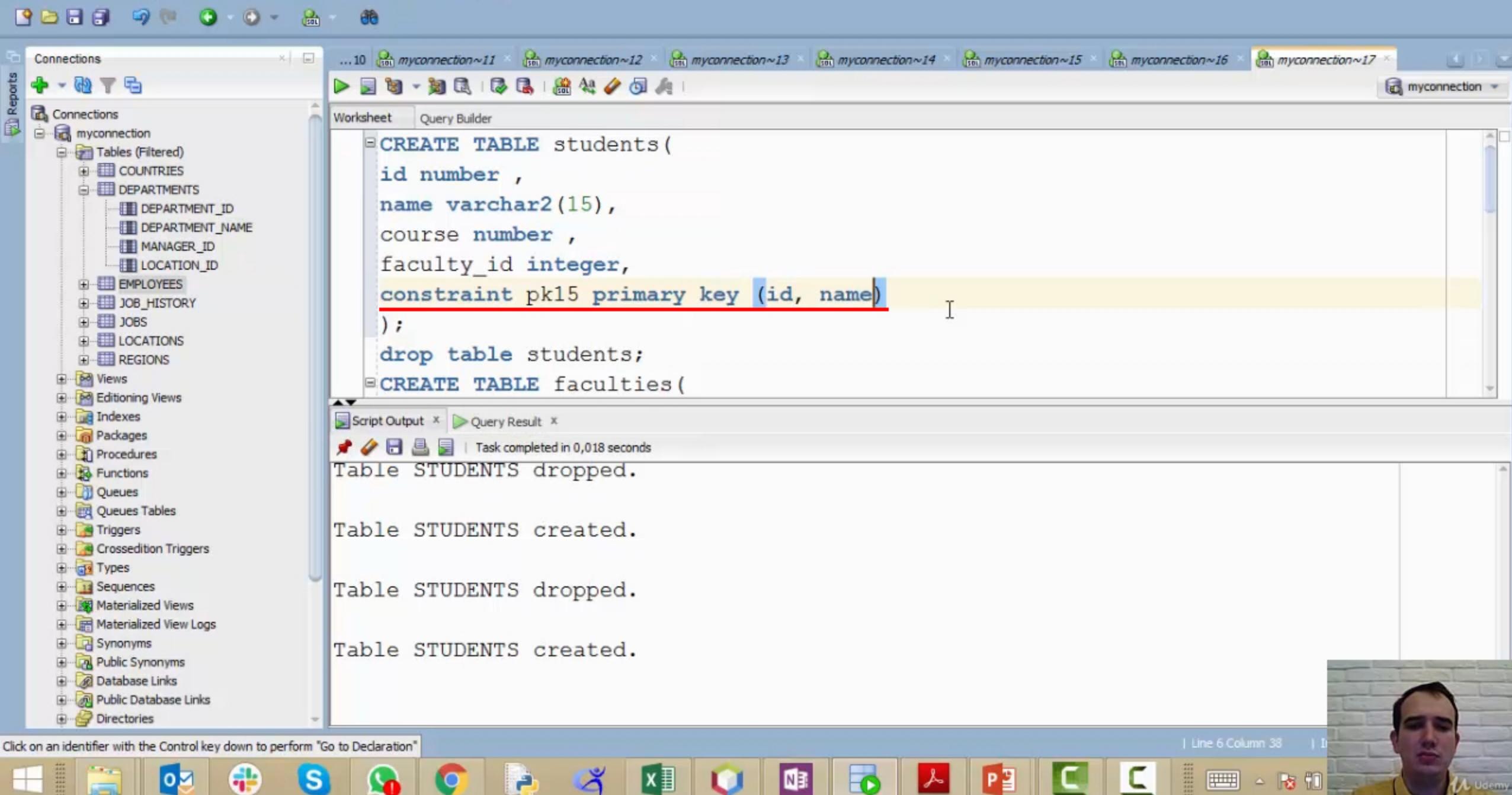
Table STUDENTS created.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 6 Column 21 | I







File Edit View Navigate Run Source Team Tools Window Help

Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
INSERT into students (id, name, faculty_id) values
(1, 'Zaur', 5);

ALTER TABLE students MODIFY (id primary key);
ALTER TABLE students MODIFY (course null);
Alter table students add constraint st_c_notnull unique (course);

insert into students values (7, 'Misha', null, 5);
insert into students values (8, 'Pasha', 3, 7);
```

Script Output x Query Result x

Task completed in 0,026 seconds

Table STUDENTS created.

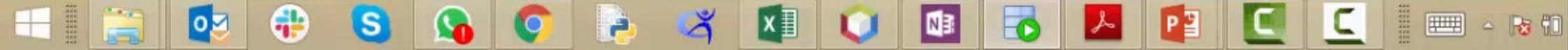
Table STUDENTS dropped.

Table STUDENTS created.

Table STUDENTS altered.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 22 Column 45 | I



Reports

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
INSERT into students (id, name, faculty_id) values
(1, 'Zaur', 5);

ALTER TABLE students MODIFY (id constraint pk primary key);
ALTER TABLE students MODIFY (course null);
Alter table students add constraint st_c_notnull unique (course);

insert into students values (7, 'Misha', null, 5);
insert into students values (8, 'Pasha', 3, 7);
```

Script Output x Query Result x

Task completed in 0,029 seconds

Table STUDENTS altered.

Table STUDENTS dropped.

Table STUDENTS created.

Table STUDENTS altered.



Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 22 Column 46 | I



Reports

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
INSERT into students (id, name, faculty_id) values
(1, 'Zaur', 5);

ALTER TABLE students MODIFY (id constraint pk primary key);
ALTER TABLE students MODIFY (course null);
Alter table students add constraint pk primary key (id);
```

```
insert into students values (7, 'Misha', null, 5);
insert into students values (8, 'Pasha', 3, 7);
```

Script Output x Query Result x
Task completed in 0,019 seconds

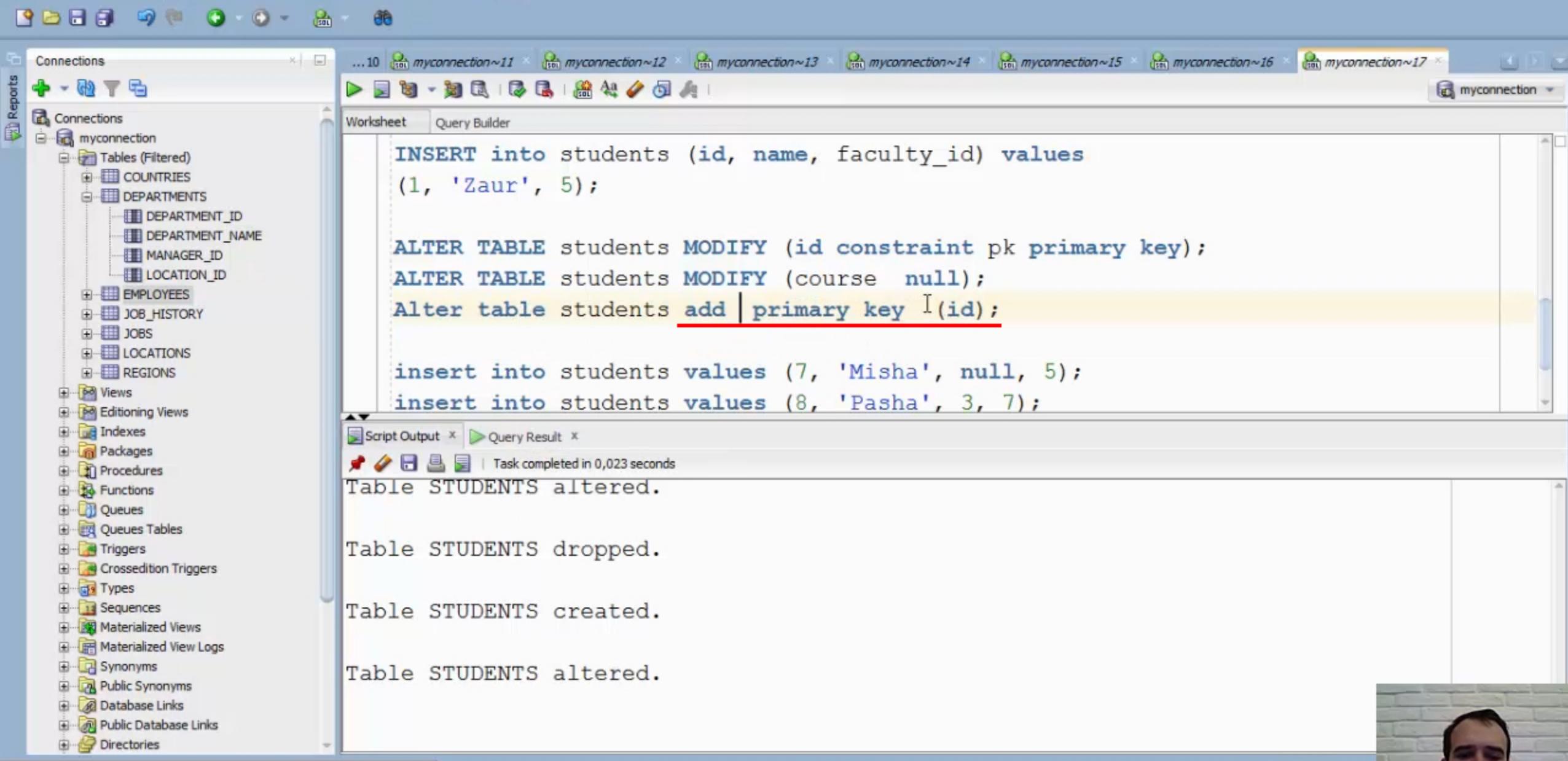
Table STUDENTS altered.

Table STUDENTS dropped.

Table STUDENTS created.

Table STUDENTS altered.





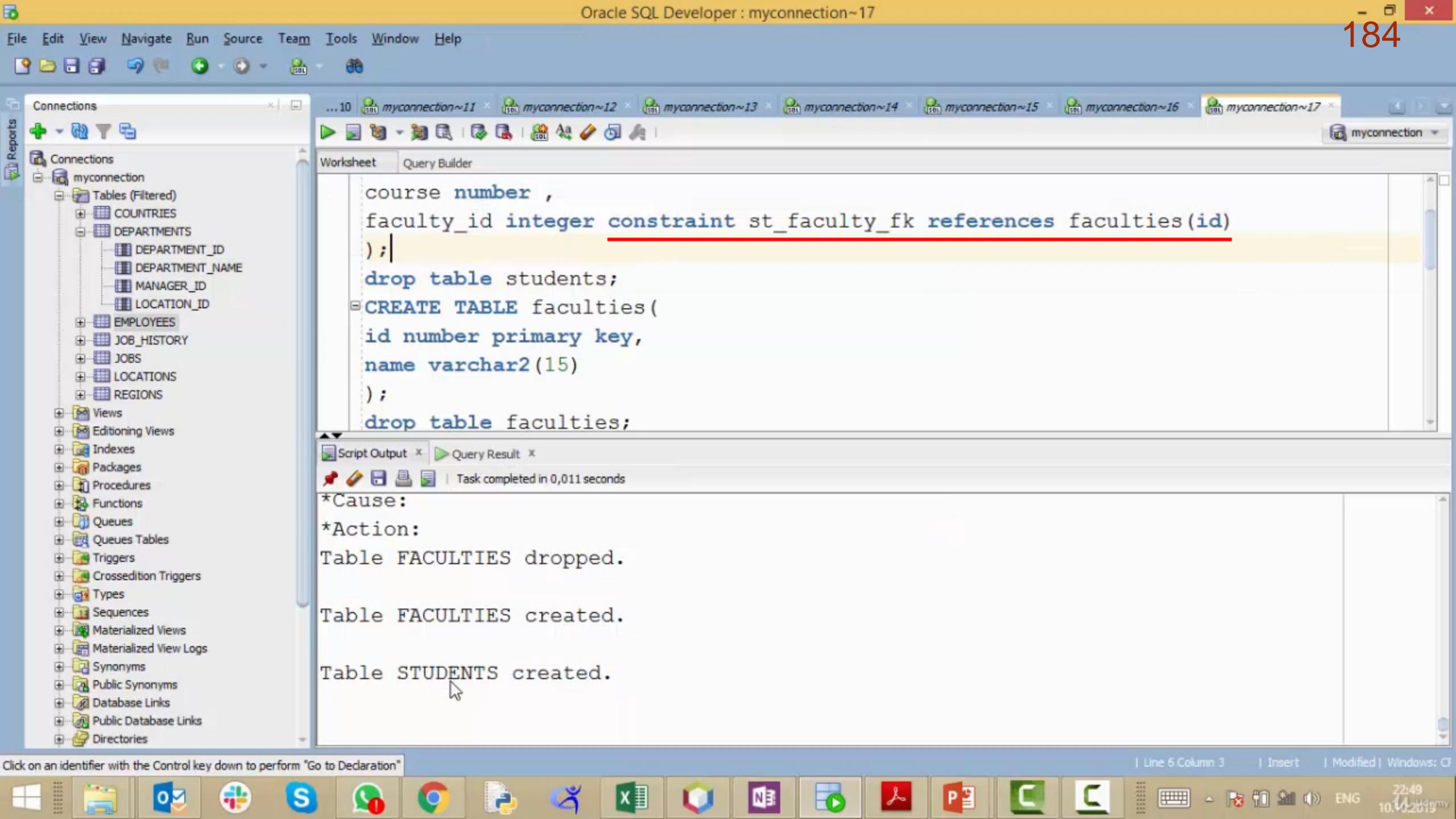
CONSTRAINT TYPES

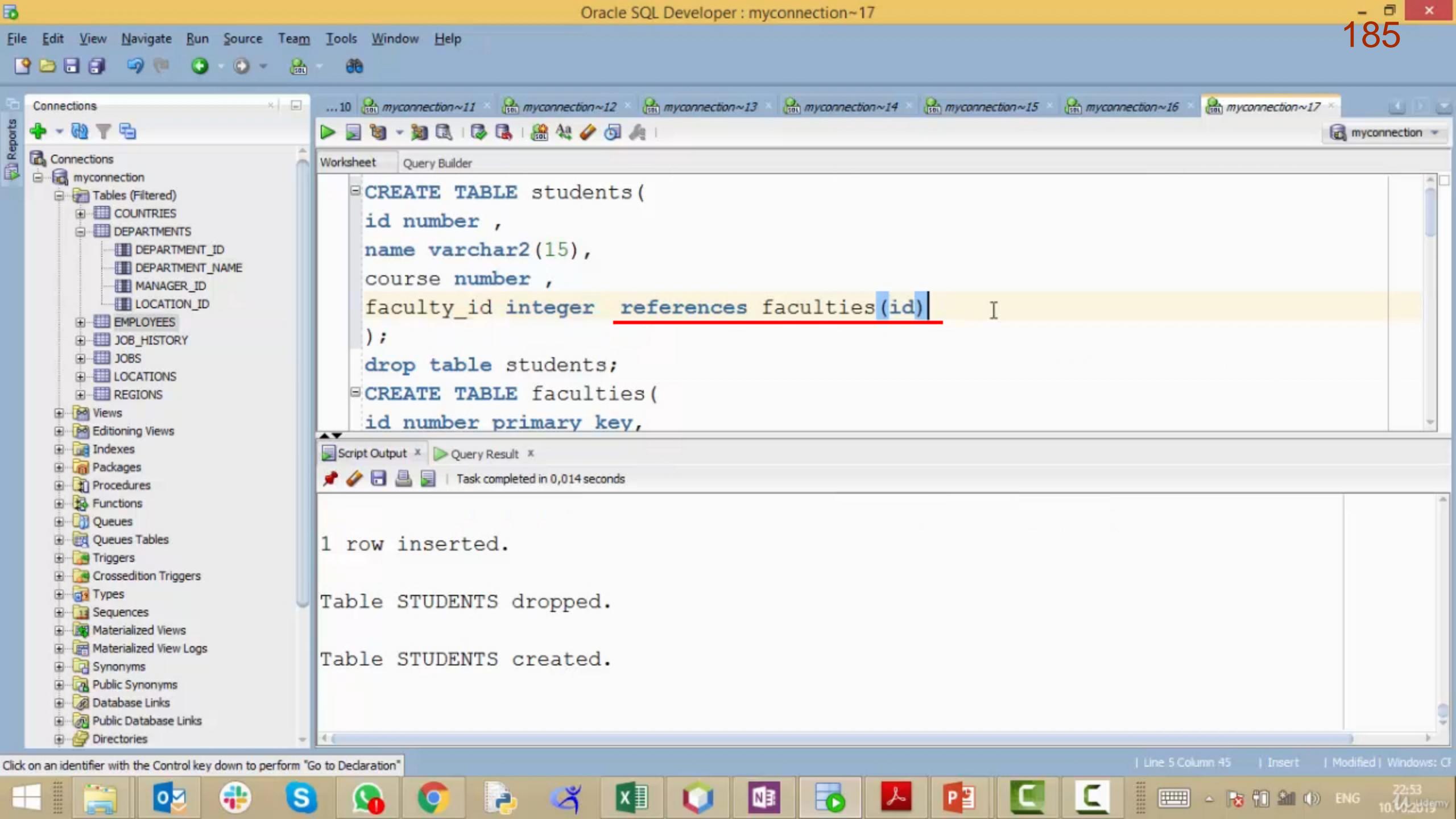
UNIQUE принуждает столбец(цы) содержать только уникальные значения. Исключение – null.

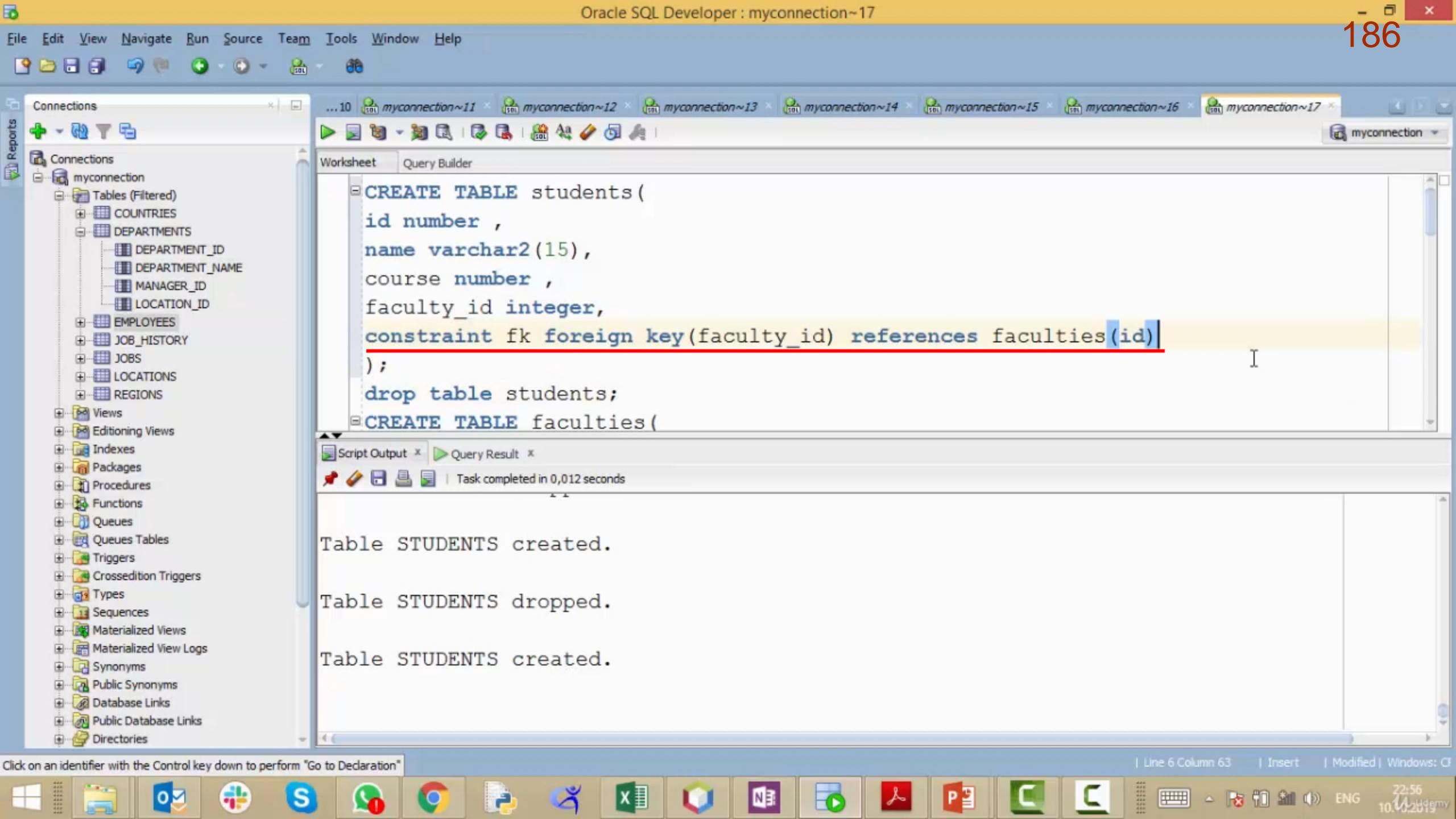
NOT NULL не разрешает столбцам содержать значение “null”

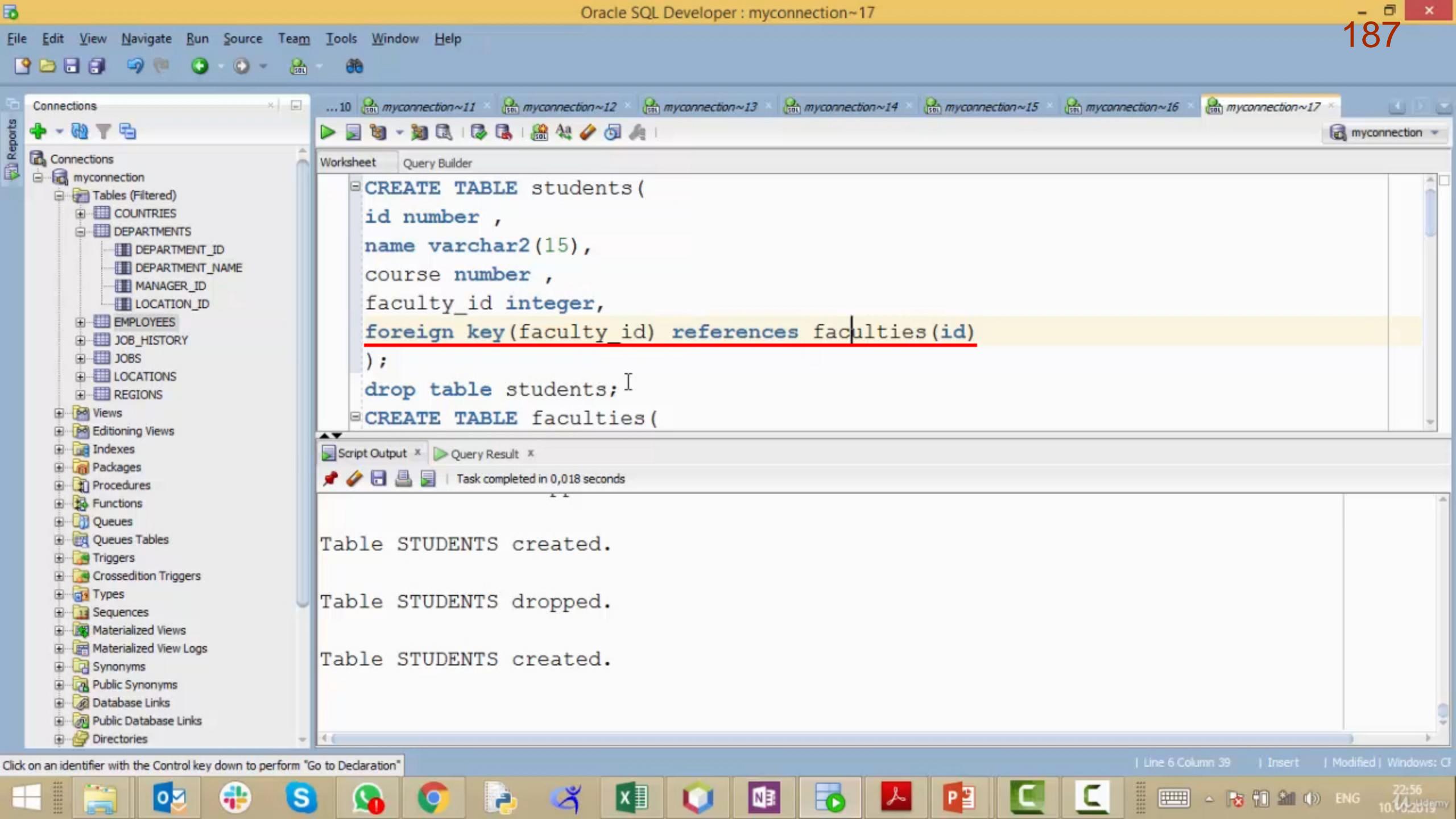
PRIMARY KEY принуждает столбец(цы) содержать только уникальные значения и не разрешает содержать значение “null”

FOREIGN KEY принуждает использовать только значения из определённого столбца таблицы-родителя или значение “null”. 









Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number ,
    name varchar2(15),
    course number ,
    faculty_id integer,
    foreign key(faculty_id) references faculties
);
drop table students;
CREATE TABLE faculties(
```

Script Output x Query Result x

Task completed in 0,02 seconds

Table STUDENTS created.

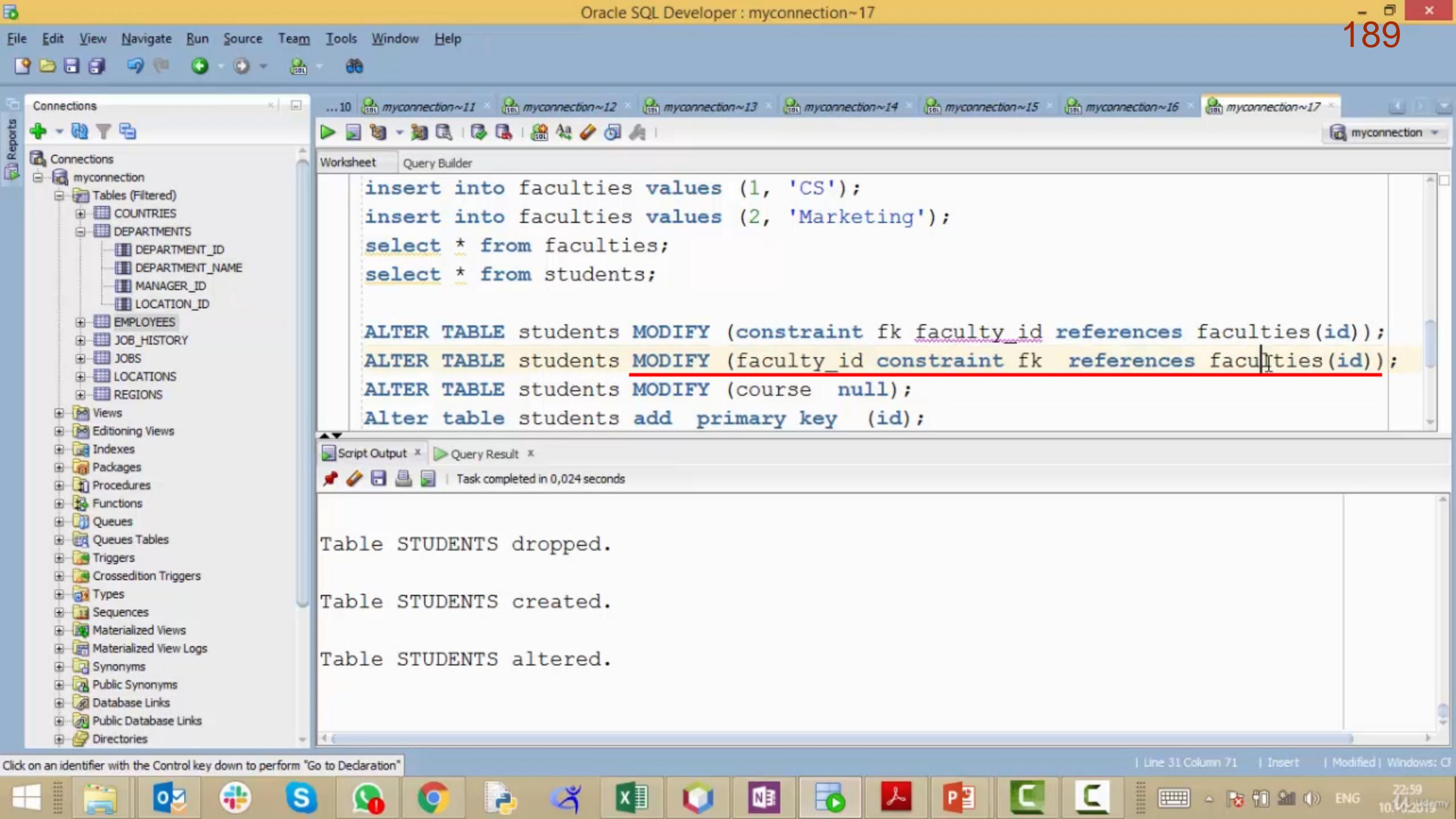
Table STUDENTS dropped.

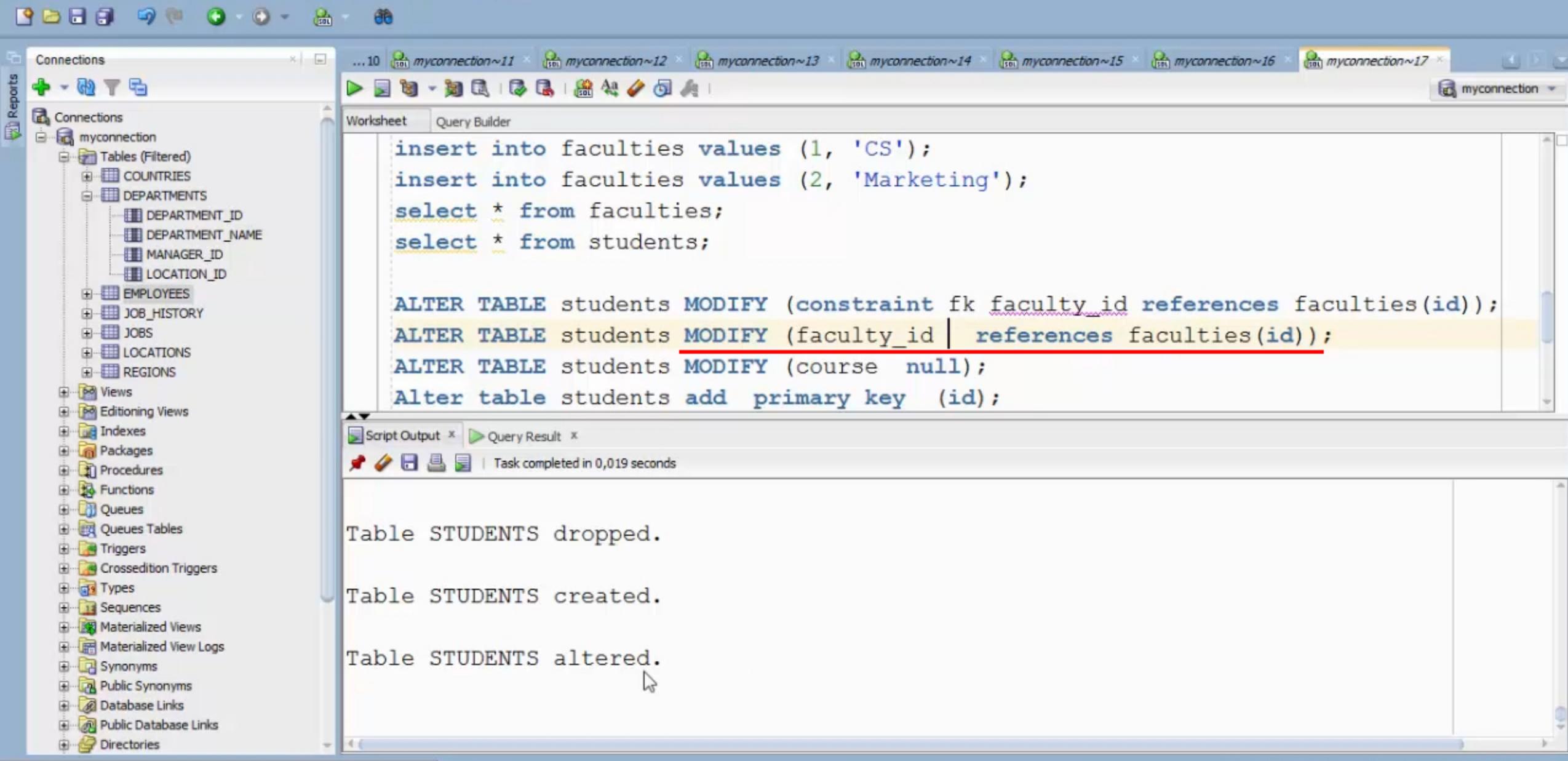
Table STUDENTS created.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 6 Column 45 | Insert | Modified | Windows: C







Click on an identifier with the Control key down to perform "Go to Declaration".





Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
ALTER TABLE students MODIFY (faculty_id references faculties(id));
ALTER TABLE students MODIFY (course null);
Alter table students add constraint fk foreign key (faculty_id)
references faculties(id);

insert into students values (7, 'Misha', null, 5);
insert into students values (8, 'Pasha', 3, 7);
insert into students (id, name, faculty_id)
values (5, 'Zaur', 3);
```

Script Output x Query Result x

Task completed in 0,017 seconds

Error report -

SQL Error: ORA-01735: invalid ALTER TABLE option
01735. 00000 - "invalid ALTER TABLE option"

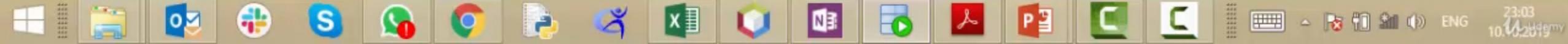
*Cause:

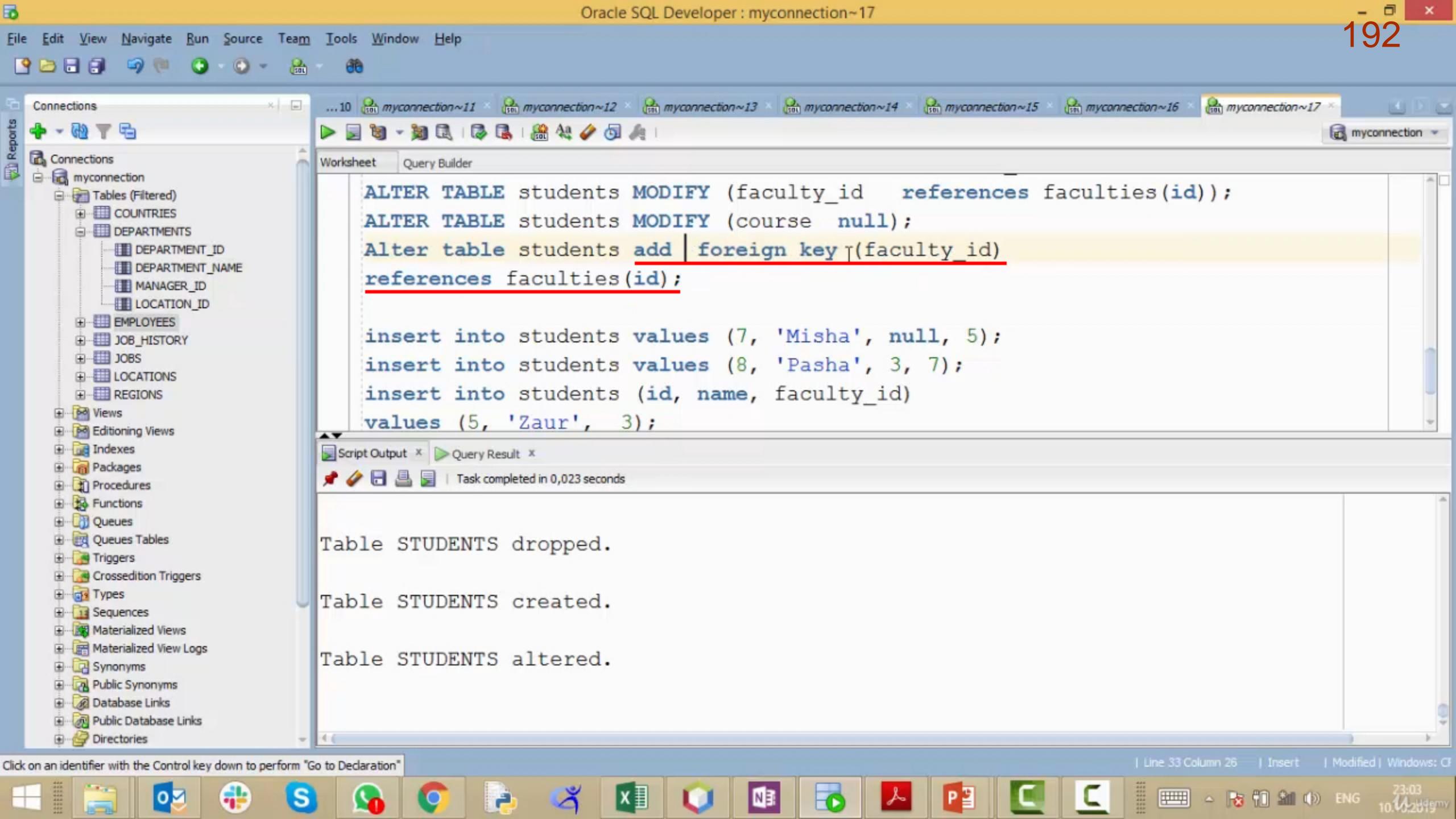
*Action:

Table STUDENTS altered.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 34 Column 25 | Insert | Modified | Windows: C





Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number ,
    name varchar2(15),
    course number ,
    faculty_id integer references faculties ON DELETE CASCADE
);
drop table students;
CREATE TABLE faculties(
    id number primary key,
```

Script Output x Query Result x

Task completed in 0,014 seconds

1 row deleted.

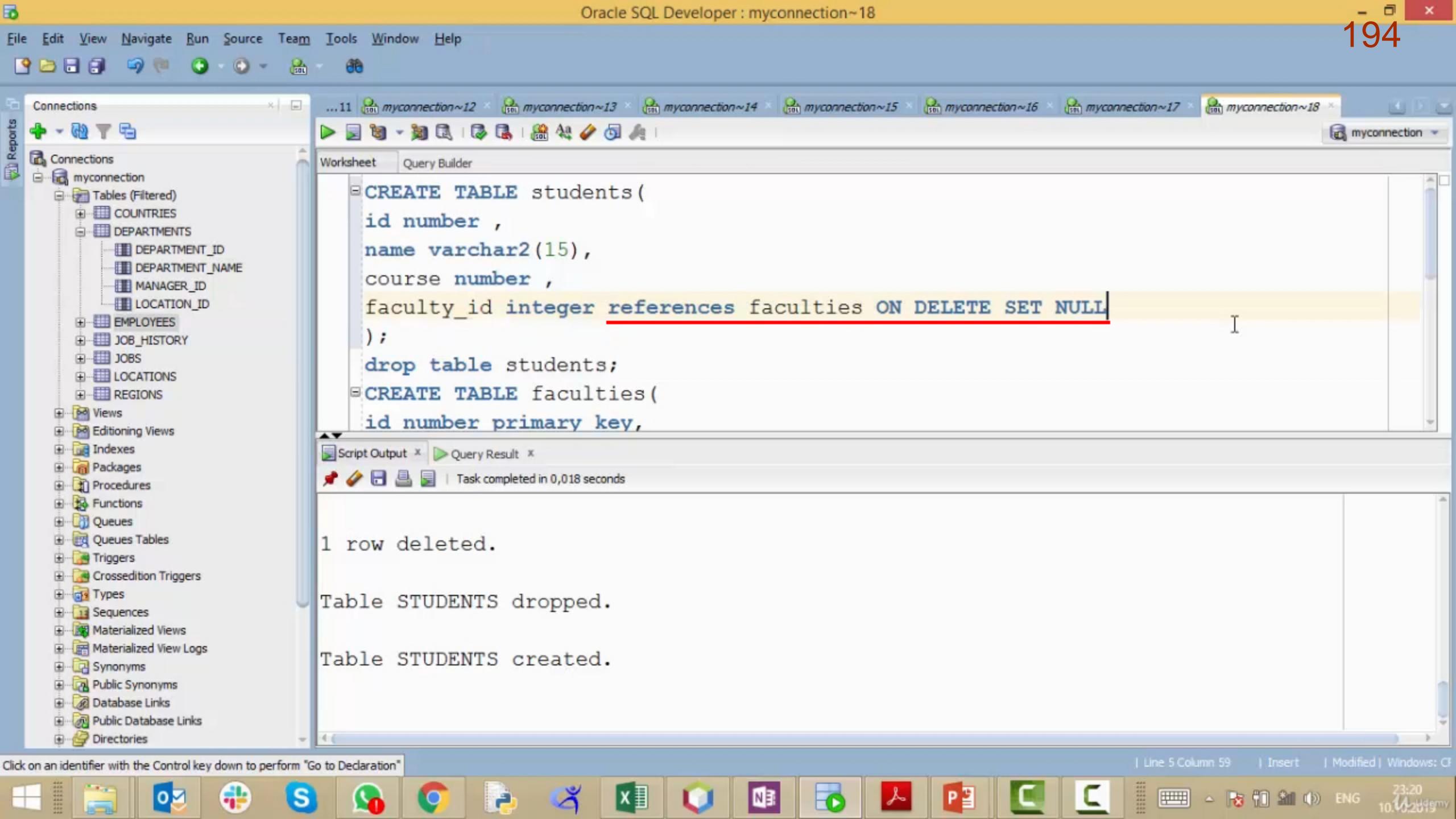
Table STUDENTS dropped.

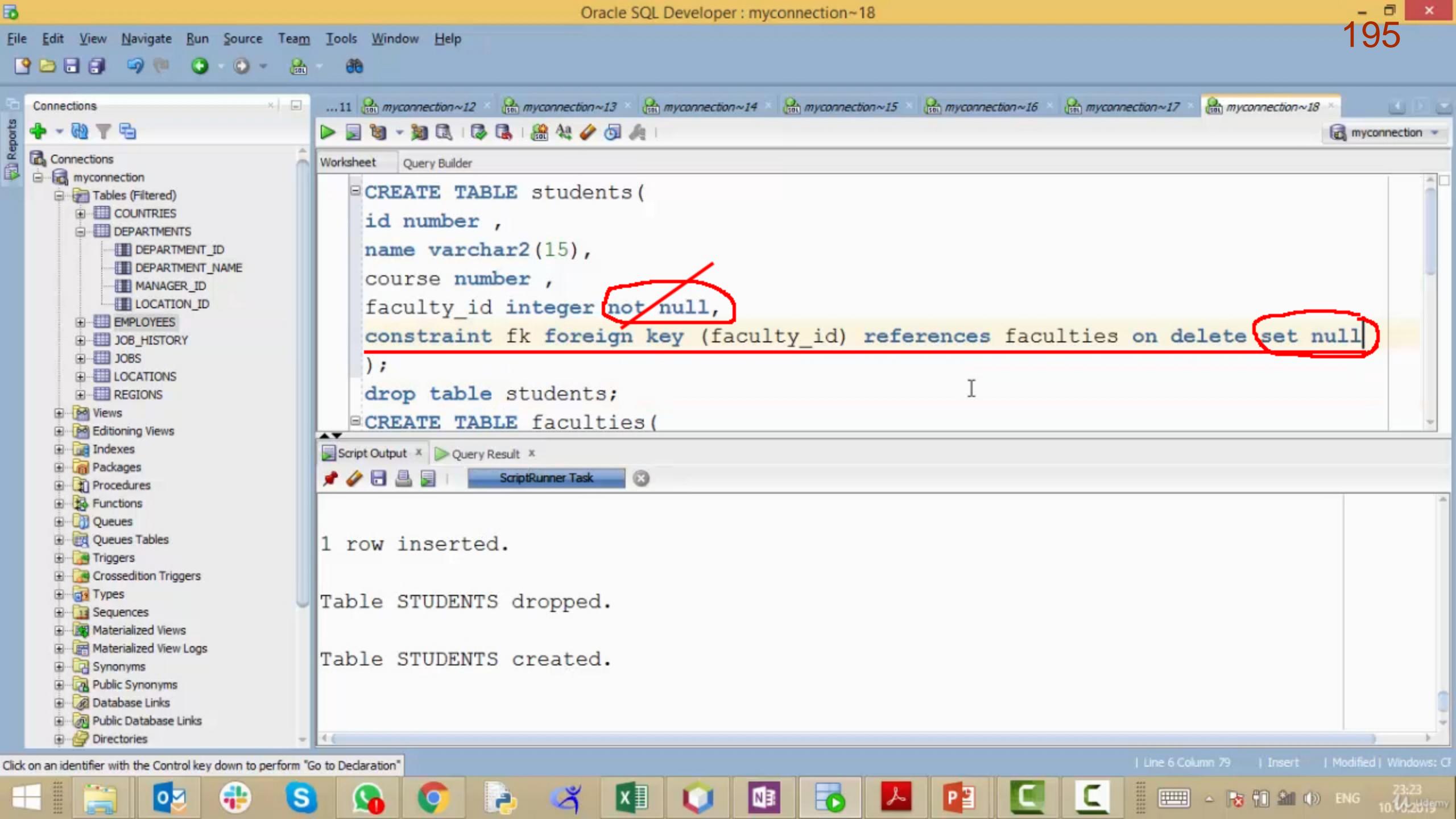
Table STUDENTS created.

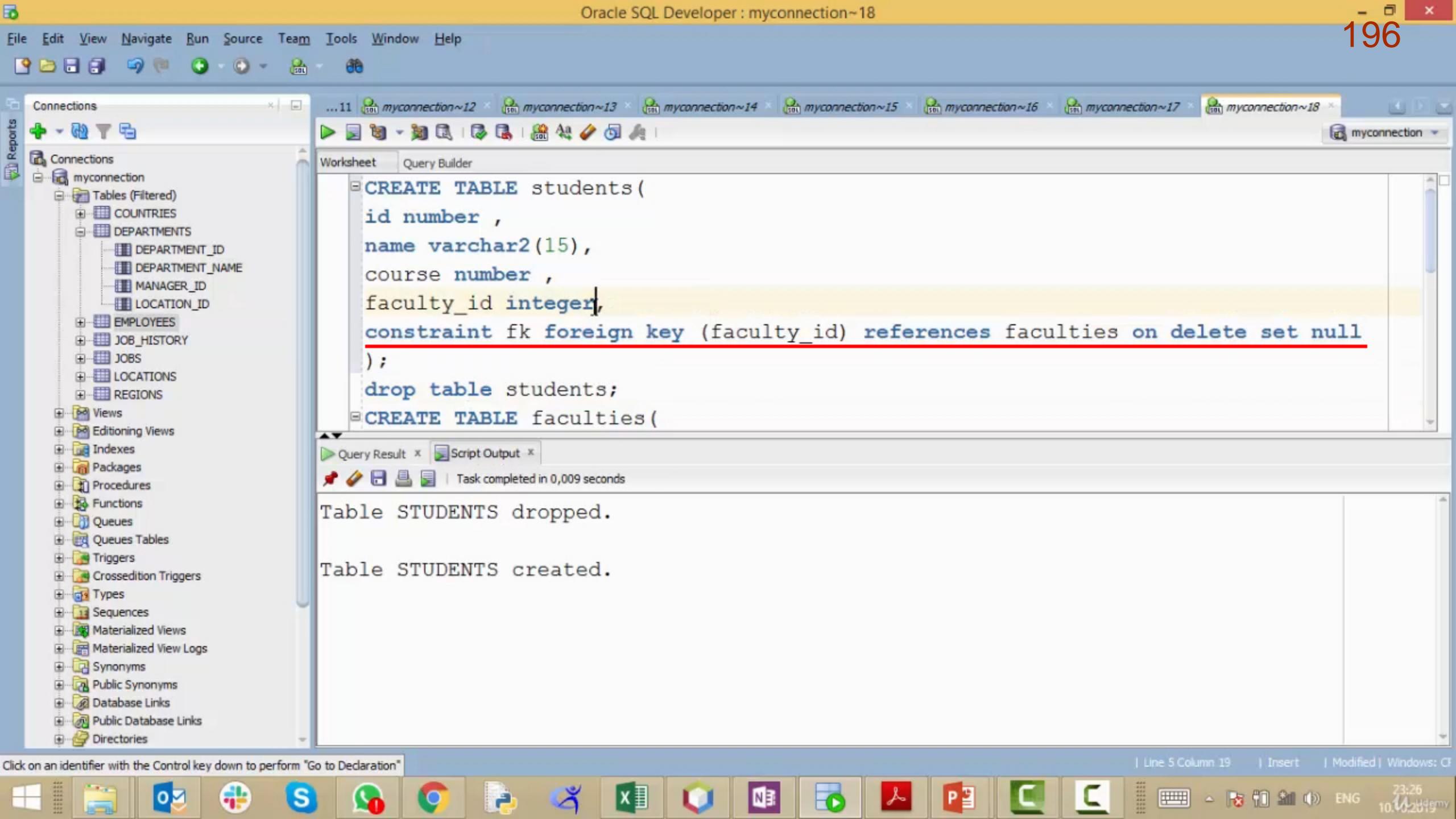
Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 5 Column 58 | Insert | Modified | Windows: C









CONSTRAINT TYPES

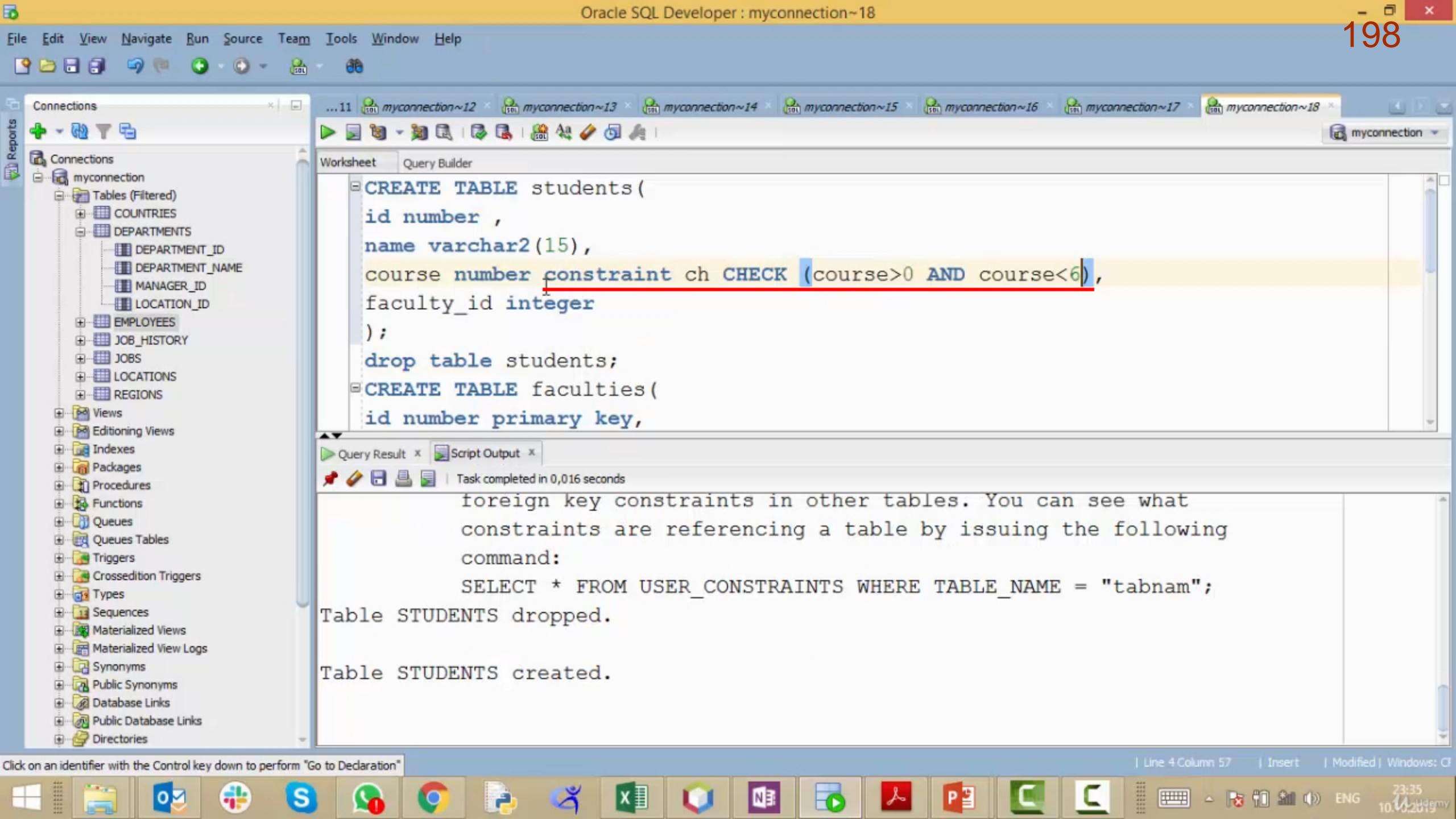
UNIQUE принуждает столбец(цы) содержать только уникальные значения. Исключение – null.

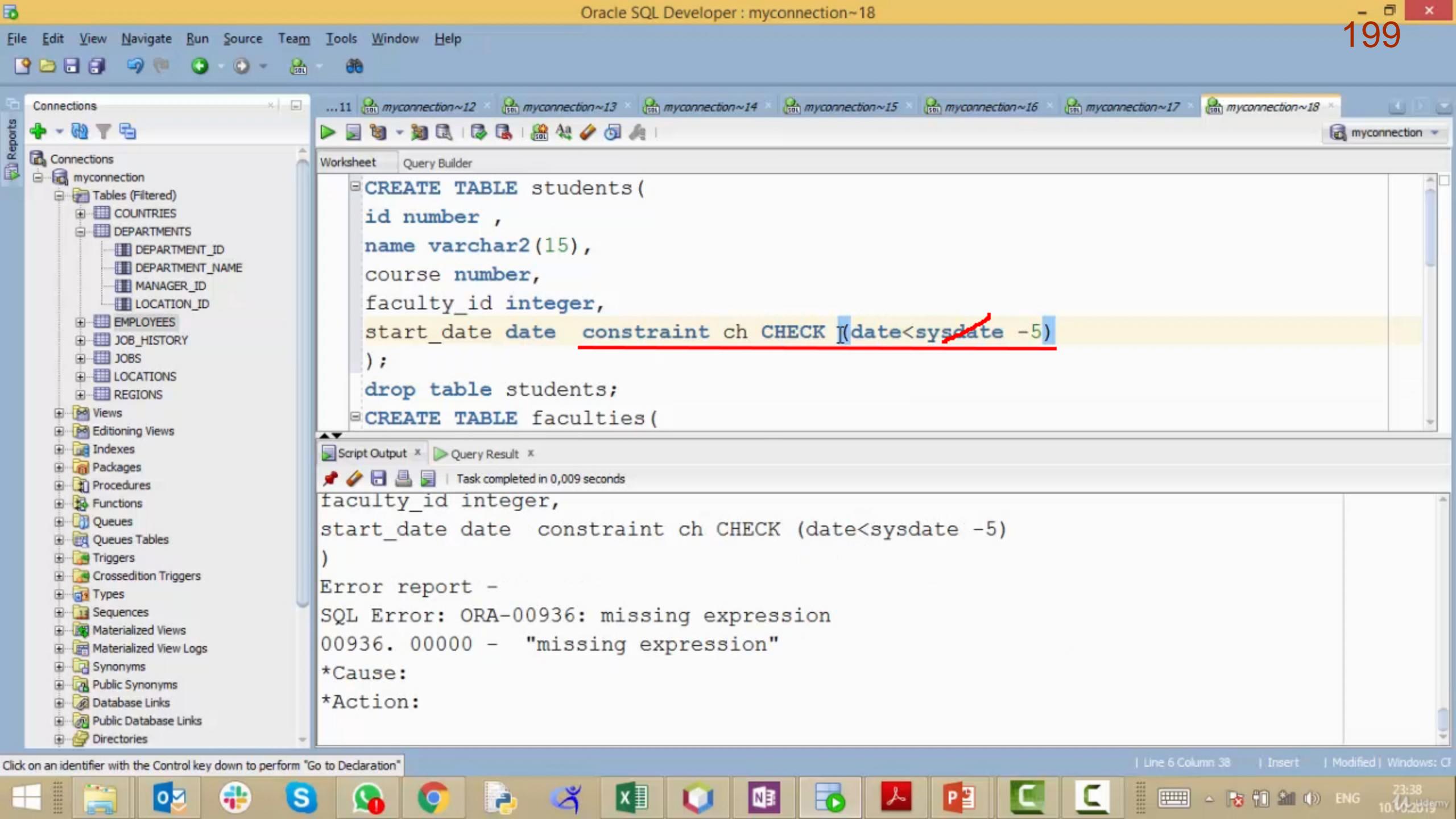
NOT NULL не разрешает столбцам содержать значение “null”

PRIMARY KEY принуждает столбец(цы) содержать только уникальные значения и не разрешает содержать значение “null”

FOREIGN KEY принуждает использовать только значения из определённого столбца таблицы-родителя или значение “null”. 

CHECK принуждает использовать только значения, которые удовлетворяют его условию(ям)







Connections

- + Connections
 - myconnection
- + Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBs
 - LOCATIONS
 - REGIONS
- + Views
- + Editioning Views
- + Indexes
- + Packages
- + Procedures
- + Functions
- + Queues
- + Queues Tables
- + Triggers
- + Crossedition Triggers
- + Types
- + Sequences
- + Materialized Views
- + Materialized View Logs
- + Synonyms
- + Public Synonyms
- + Database Links
- + Public Database Links
- + Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number ,
    name varchar2(15),
    course number constraint ch CHECK (course >id),
    faculty_id integer
);
drop table students;
CREATE TABLE faculties(
    id number primary key,
    faculty_id integer
)
Error report -
SQL Error: ORA-02438: Column check constraint cannot reference other columns
02438. 00000 - "Column check constraint cannot reference other columns"
*Cause: attempted to define a column check constraint that references
another column.
*Action: define it as a table check constraint.
```

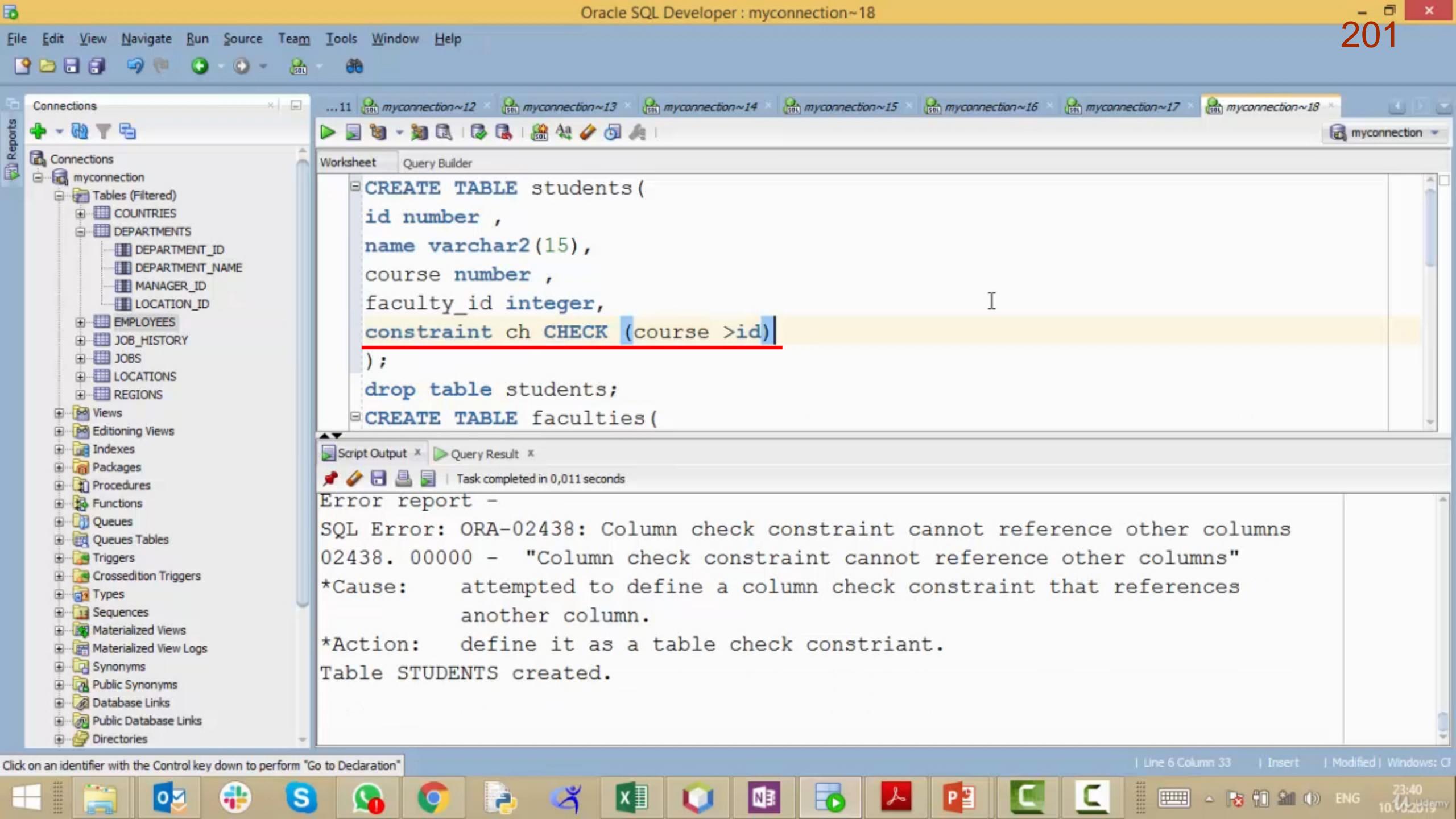
Script Output x Query Result x

Task completed in 0,01 seconds

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 4 Column 46 | Insert | Modified | Windows: C





Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number ,
    name varchar2(15),
    course number ,
    faculty_id integer,
    CHECK (course >id)
);
drop table Students;
CREATE TABLE faculties(
```

Script Output x Query Result x

Task completed in 0,008 seconds

another column.

*Action: define it as a table check constraint.

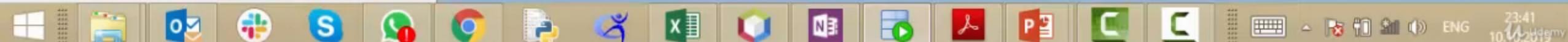
Table STUDENTS created.

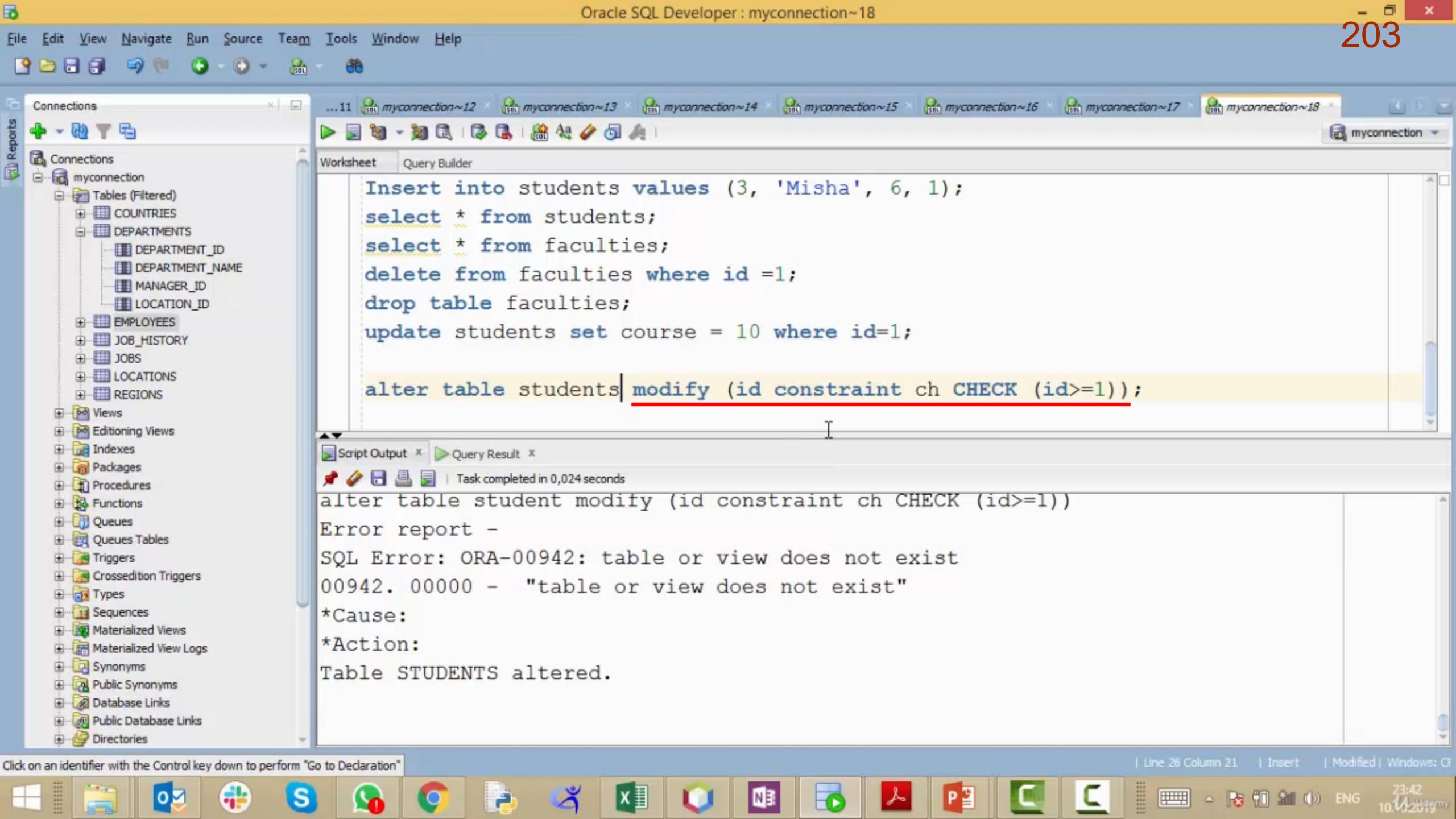
Table STUDENTS dropped.

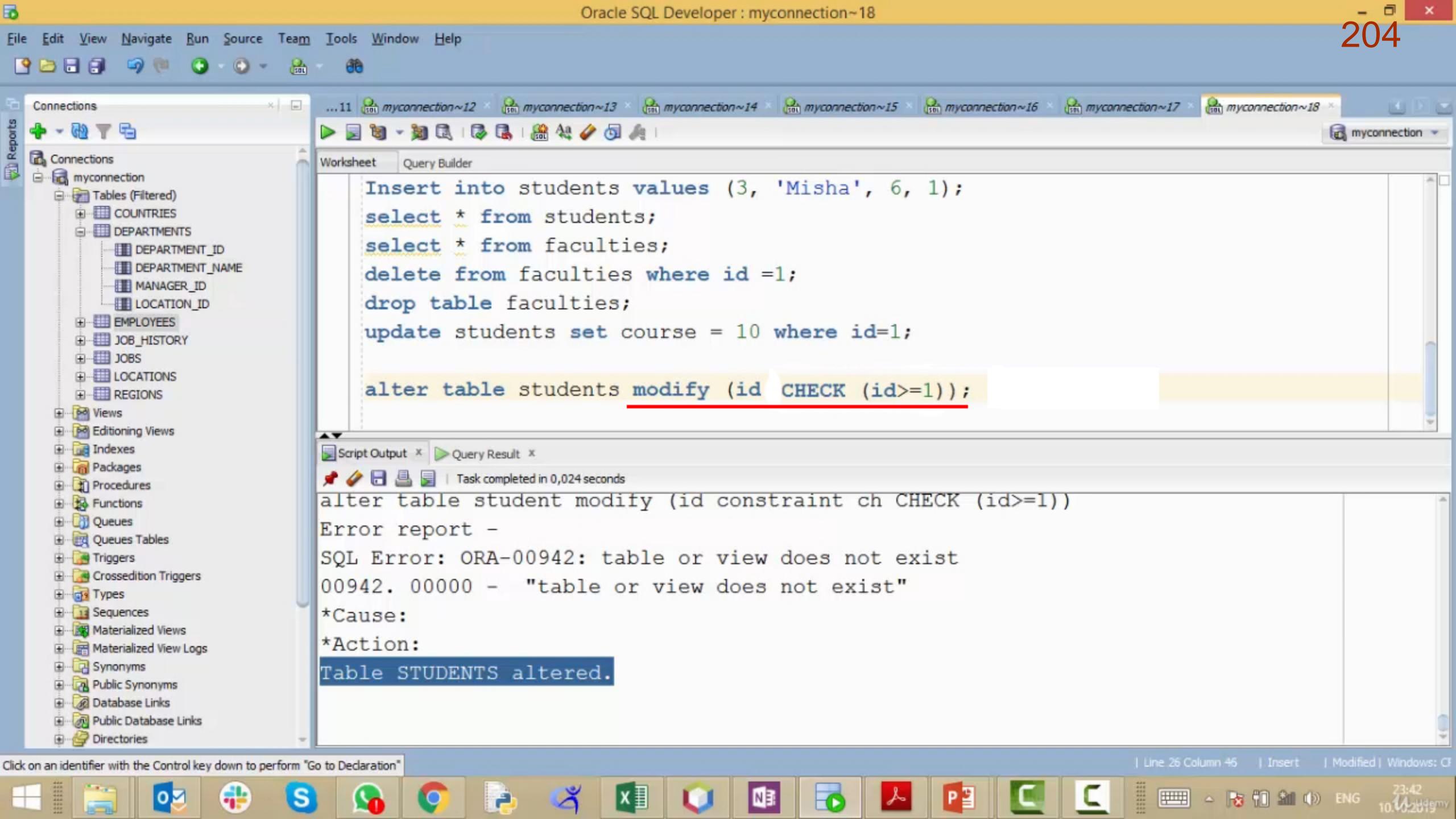
Table STUDENTS created.

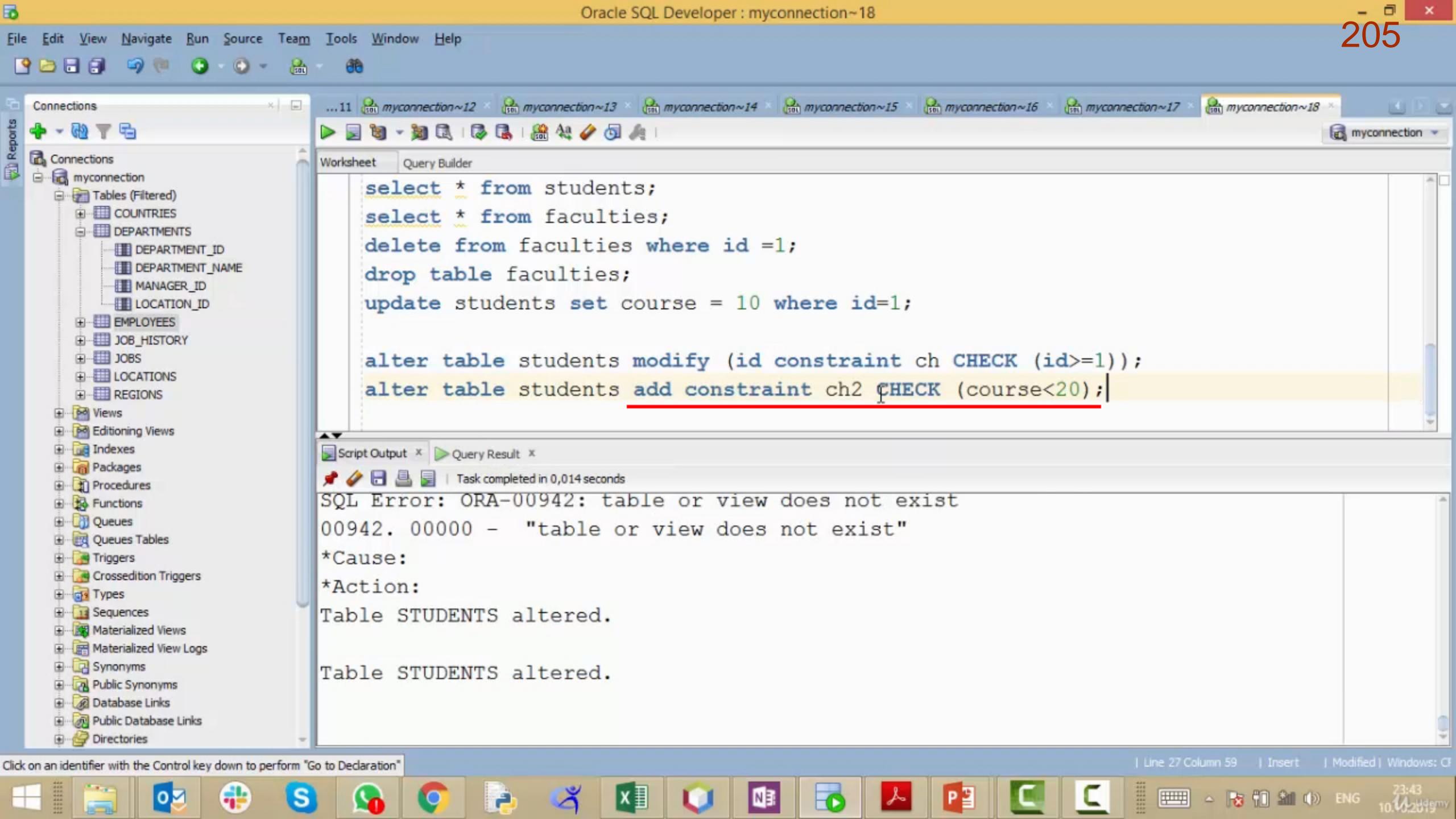
Click on an identifier with the Control key down to perform "Go to Declaration"

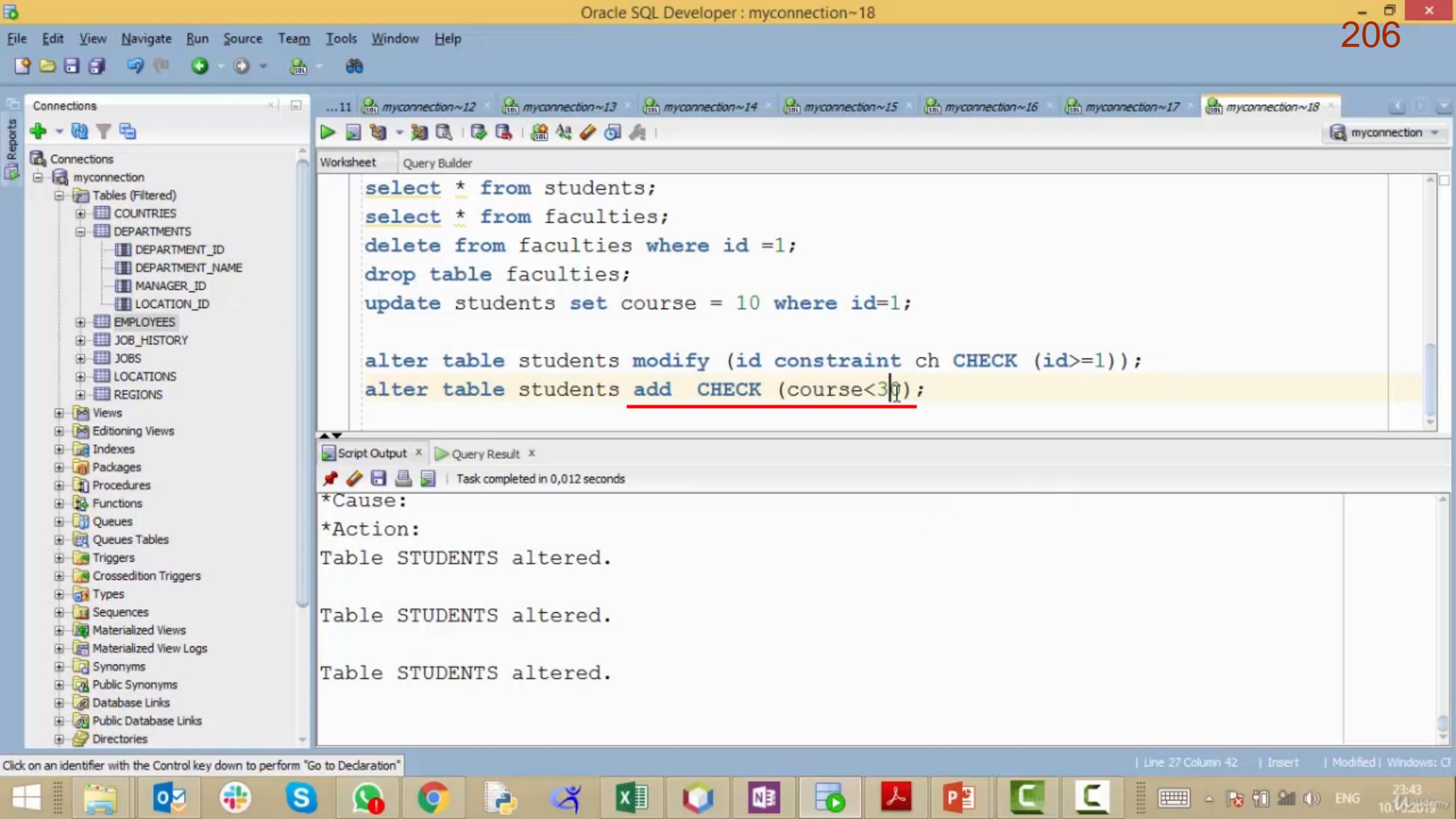
| Line 6 Column 1 | Insert | Modified | Windows: C











File Edit View Navigate Run Source Team Tools Window Help

Connections

+ myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editing Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Worksheet Query Builder

```
CREATE TABLE students(
    id number ,
    name varchar2(15),
    course number ,
    email varchar(15) CHECK (INSTR(email, '@')>0) UNIQUE,
    faculty_id integer
);
drop table students;
CREATE TABLE faculties(
```

Script Output x Query Result x

| Task completed in 0,026 seconds

*Action: do not insert values that violate the constraint.

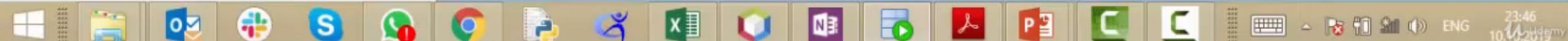
1 row inserted.

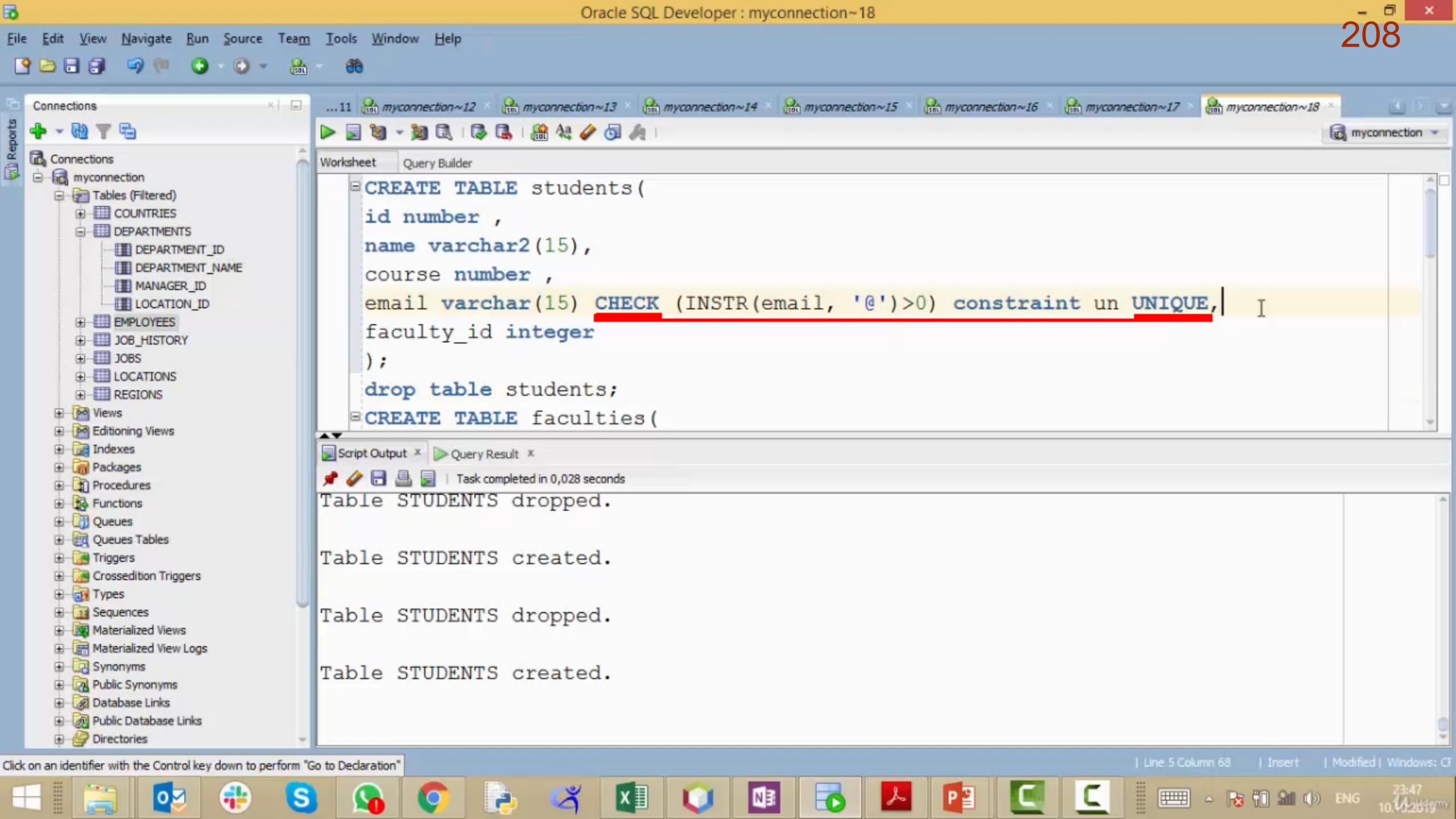
Table STUDENTS dropped.

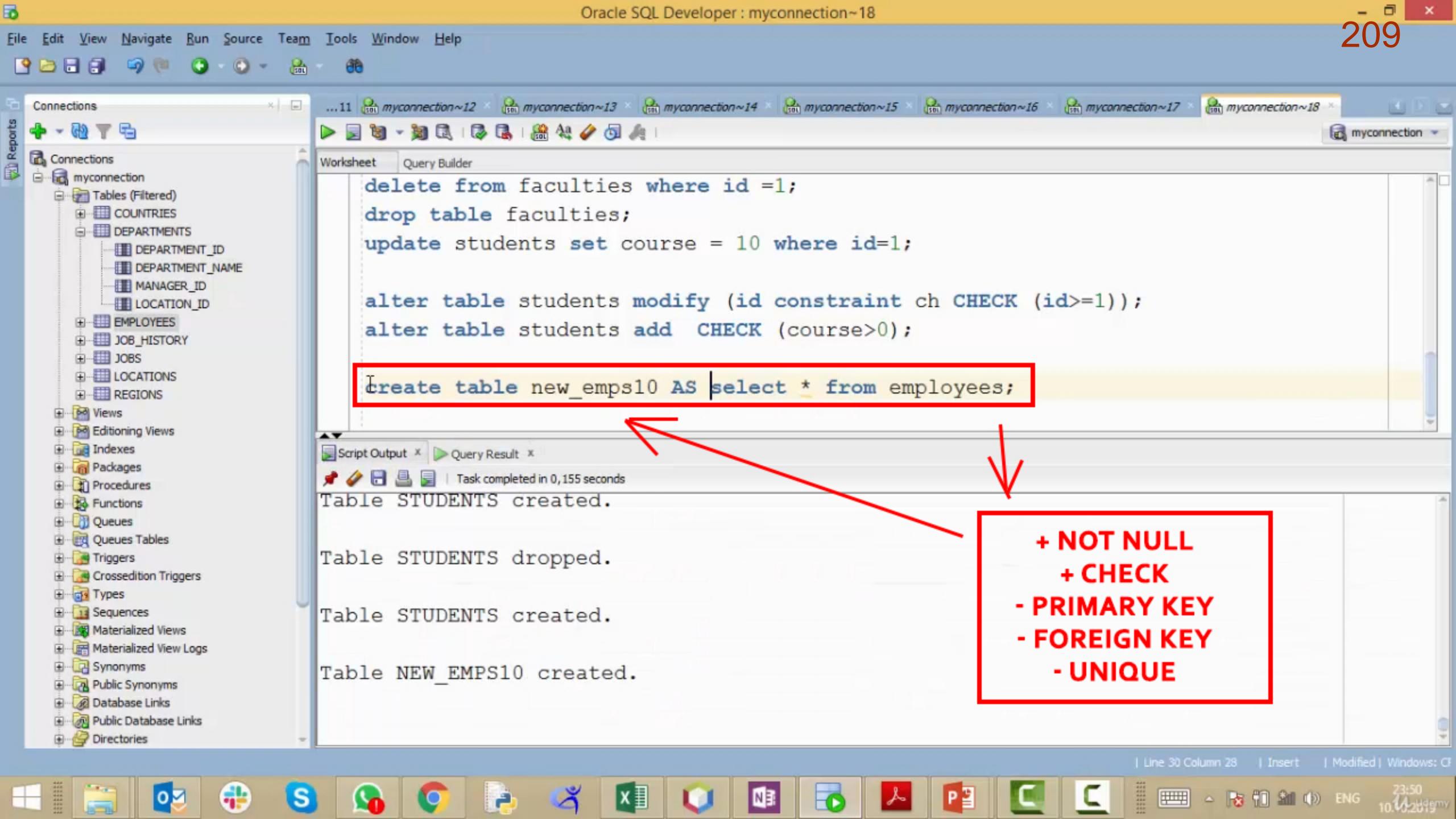
Table STUDENTS created. ↵

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 5 Column 53 | Insert | Modified | Windows: C





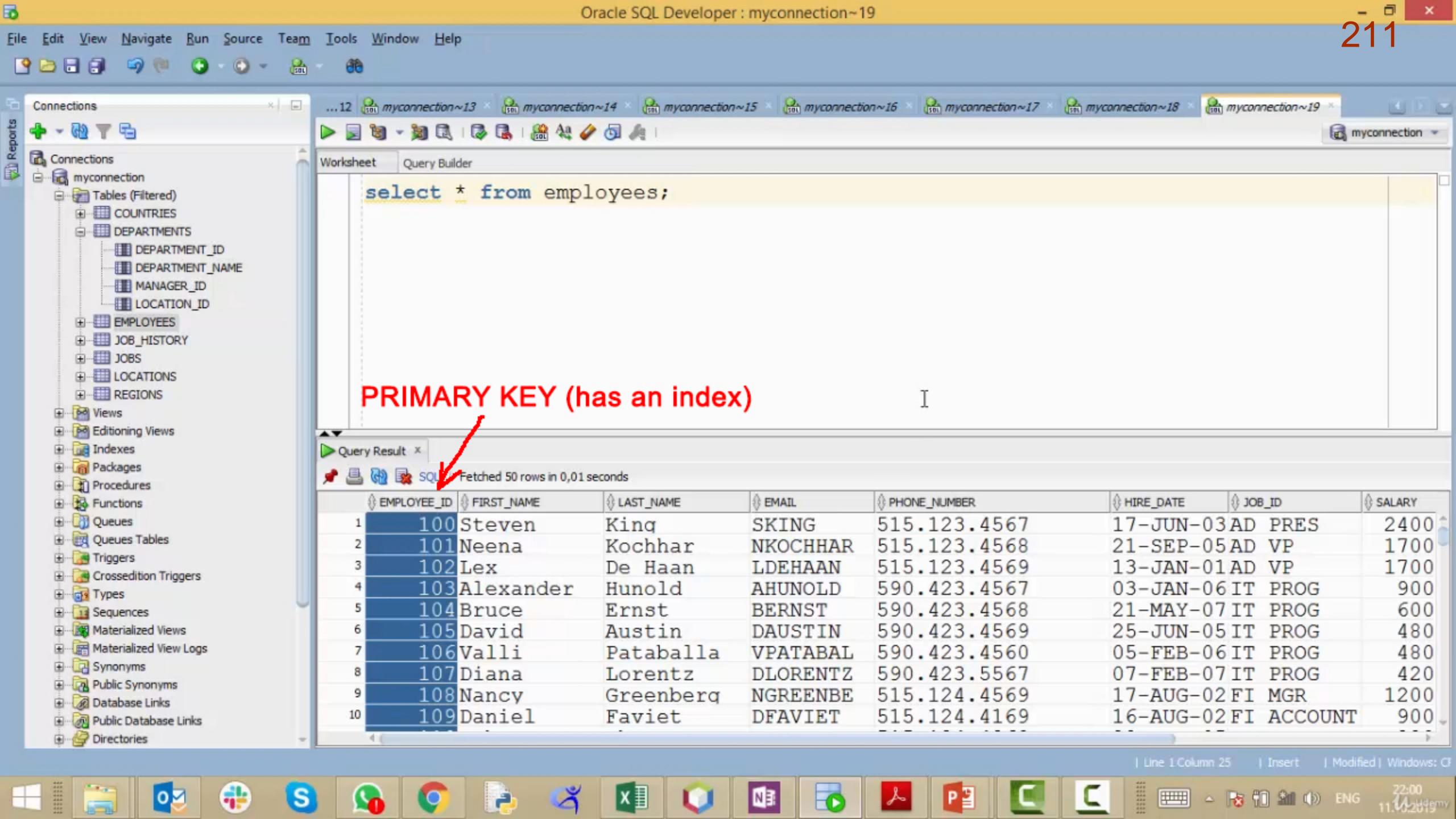


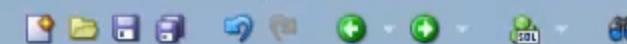
ТИПЫ ИНДЕКСОВ

Для чего нужны индексы:

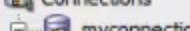
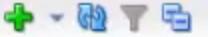
- 1) быстрая проверка PRIMARY KEY и UNIQUE (обязателен)
- 2) ускоренный поиск информации







Connections



Connections

myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editioning Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

... 12 myconnection~13 myconnection~14 myconnection~15 myconnection~16 myconnection~17 myconnection~18 myconnection~19


Worksheet

```
select * from employees;
```

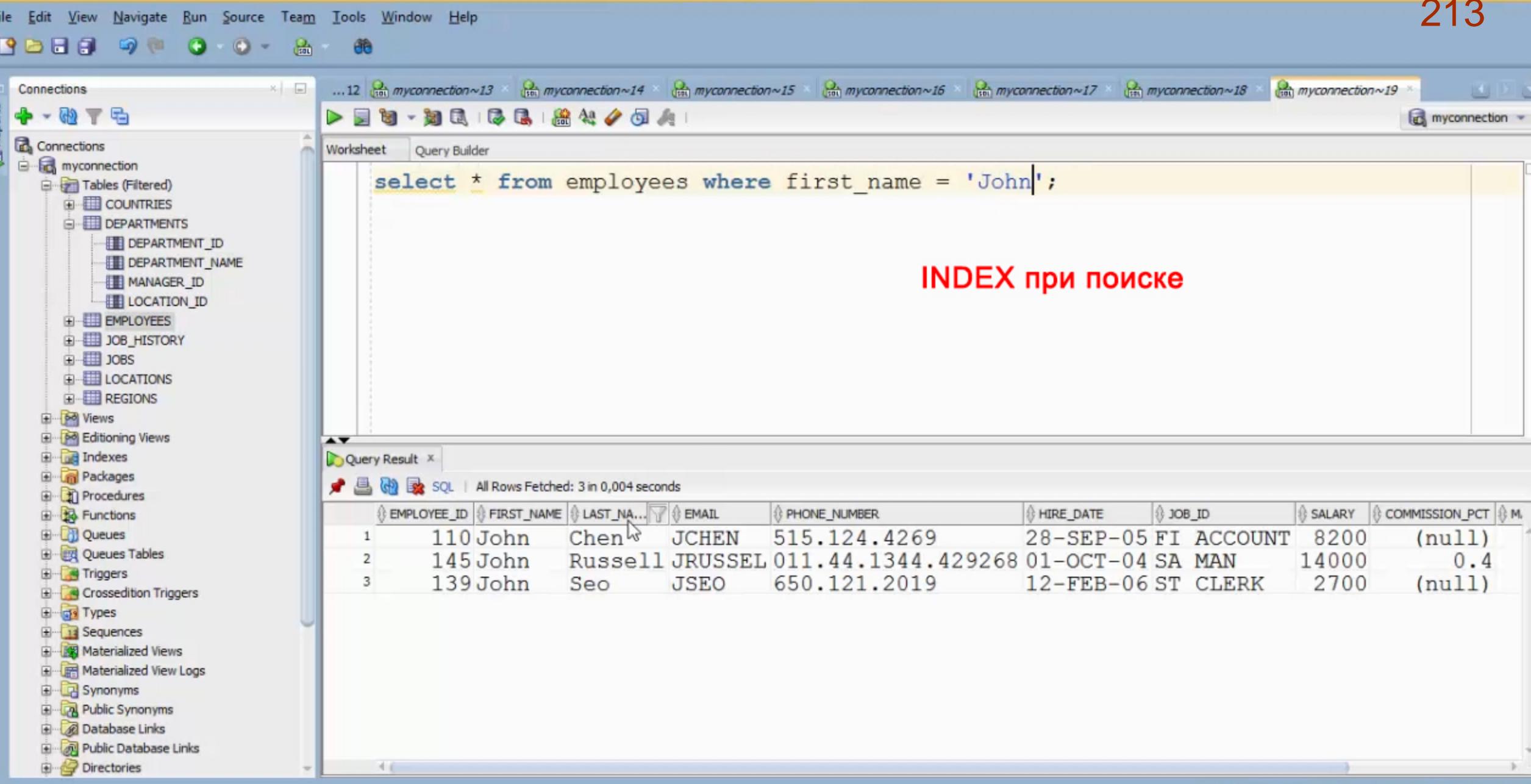
PRIMARY KEY (has an index)

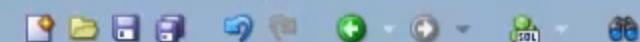
FOREIGN KEY

Query Result

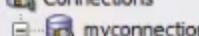
SQL | Fetched 100 rows in 0,013 seconds

	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	SKING	515.123.4567	17-JUN-03	AD PRES	24000	(null)	(null)	90
2	NKOCHHAR	515.123.4568	21-SEP-05	AD VP	17000	(null)	100	90
3	LDEHAAN	515.123.4569	13-JAN-01	AD VP	17000	(null)	100	90
4	AHUNOLD	590.423.4567	03-JAN-06	IT PROG	9000	(null)	102	60
5	BERNST	590.423.4568	21-MAY-07	IT PROG	6000	(null)	103	60
6	DAUSTIN	590.423.4569	25-JUN-05	IT PROG	4800	(null)	103	60
7 la	VPATABAL	590.423.4560	05-FEB-06	IT PROG	4800	(null)	103	60
8	DLORENTZ	590.423.5567	07-FEB-07	IT PROG	4200	(null)	103	60
9 rq	NGREENBE	515.124.4569	17-AUG-02	FI MGR	12008	(null)	101	100
10	DFAVIET	515.124.4169	16-AUG-02	FI ACCOUNT	9000	(null)	108	100





Connections



... 12 myconnection~13 myconnection~14 myconnection~15 myconnection~16 myconnection~17 myconnection~18 myconnection~19

Worksheet Query Builder

```
select * from employees where first_name = 'John';
order by
group by
union
join
```

INDEX при сортировке

INDEX при JOIN

Query Result x

All Rows Fetched: 3 in 0,004 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	M
1	110	John	Chen	JCHEN 515.124.4269	28-SEP-05	FI ACCOUNT	8200	(null)	
2	145	John	Russell	JRUSSEL 011.44.1344.429268	01-OCT-04	SA MAN	14000	0.4	
3	139	John	Seo	JSEO 650.121.2019	12-FEB-06	ST CLERK	2700	(null)	

ТИПЫ ИНДЕКСОВ



B-TREE



BITMAP



11 октября 2019 г.

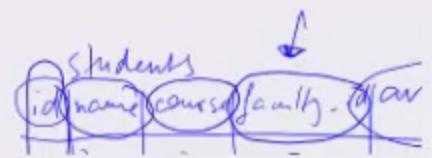
+ Страница



B-Tree

- 1) много строк

Записная книжка... > Заметки на полях



primary key [INT]
unique
~~foreign~~ select

extent 1
extent 2

1) 1-30 columns
new-emp -
a - b - ?

USERS SCHEM
username
pwd

B-Tree

1) много строк

2) комп. строк в альфабетном порядке

— максимальное

2-4 %



select * from names where name > 'B';

X

3) много узких деревьев

✓

4) where , JOIN

I

ТИПЫ ИНДЕКСОВ



B-TREE



BITMAP

```
CREATE { UNIQUE | BITMAP } INDEX  
    schema.index_name ON  
    schema.table_name (column1, column2, ...);
```

```
DROP INDEX schema.index_name;
```

+ Страница



↓ students ↓ ↓
 id name course faculty_id sex

Записная книжка... > Заметки на полях

~~Binary~~
 1) много строк
 2) компримация

~~students~~
 id name course faculty_id sex

primary key [INT]
 unique
 faculty_id
 select

extent 1
 extent 2

1) 1-30 символов
 new-emp -
 $a - b = ?$

11 октября 2019 г. 23:06



↓ students ↓ ↓ ↓

	id	name	course	faculty_id	sex
1	Ivan	3		1	M
2	Mark	4		1	F
3	Ivan	1		2	M
4	Svetlana	2		null	F
5	Petya	3		2	M

bitmap



M → 10101

F → 01010

1 → 11000

2 → 00101

null → 00010

f_id = 2 AND Sex = M

10101

00101

00101

☰ Последние

+ Страница

↓ students ↓
 id name course faculty_id sex

Записная книжка... > Заметки на полях

Все

- 1) много строк
- 2) count > n & less than

↓ students ↓
 id name course faculty_id sex

primary key (INT)
 unique
 faculty

select

extent1 ← (≡)
 extent2

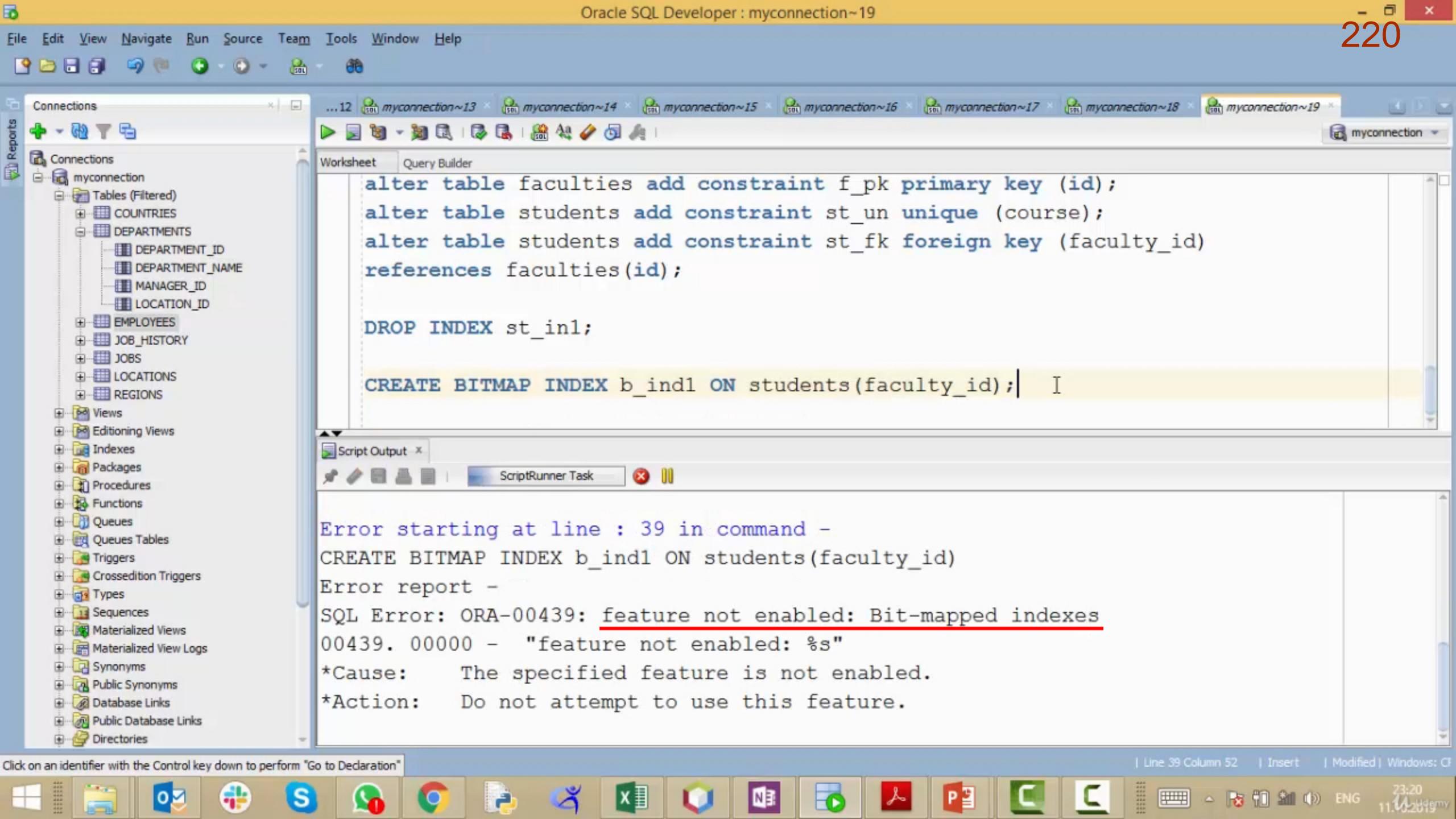
1) 1-30 символов
 new-emp -
 $a - b = ?$

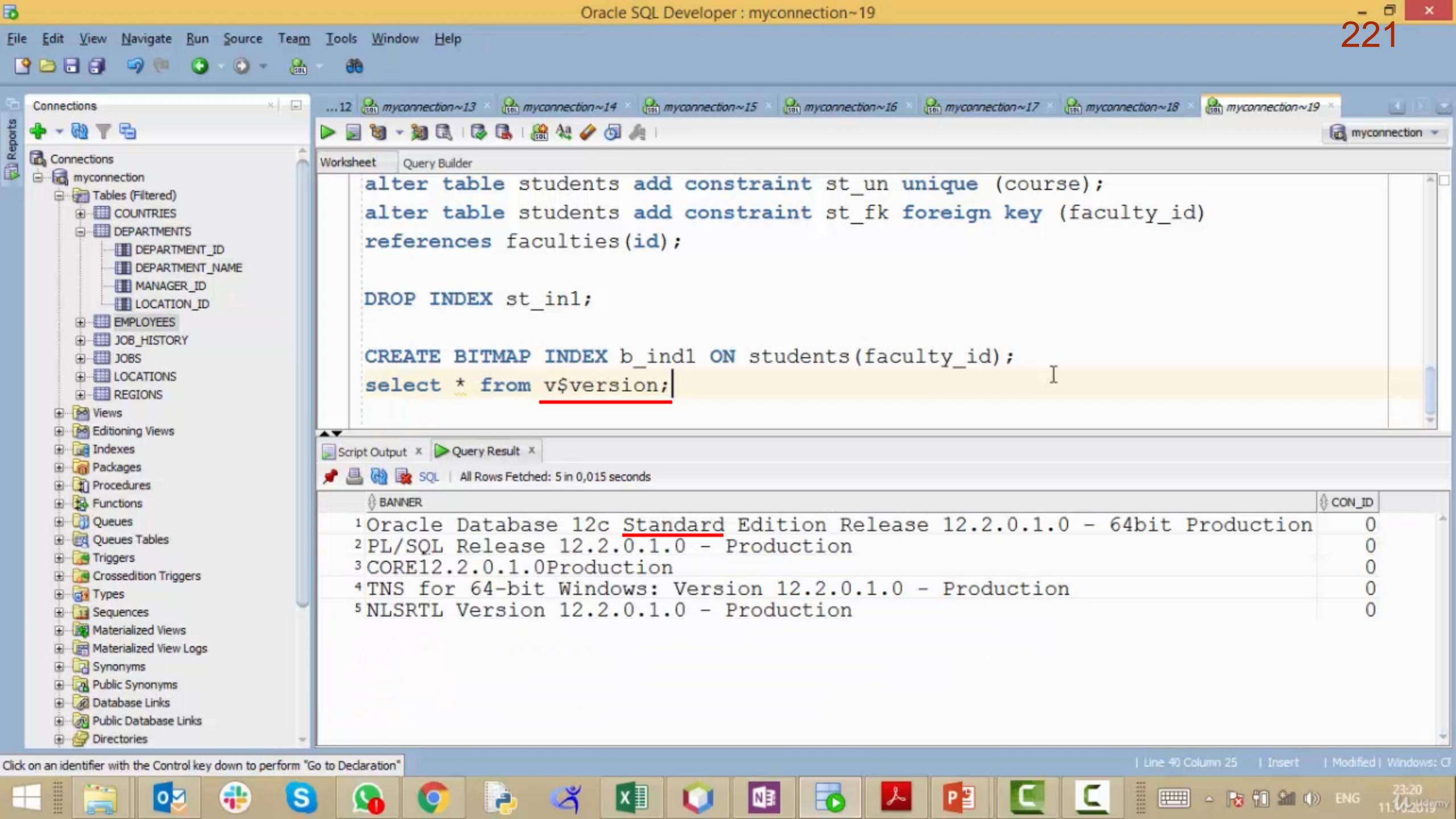
	id	name	course	faculty_id	sex
1	Ivan	3	1	M	
2	Masha	4	1	F	
3	Ivan	1	②	M	
4	Svetlana	2	null	F	
5	Petya	3	②	M	

faculty_id = 2 AND Sex = M

1) make non-unique query
 2) count. open ↑
 3) AND, OR, NOT.

10101
 00101
 00101





Причины использования:

1. Конфиденциальность
2. Безопасность
3. Простота для пользователей
4. Названия таблиц или столбцов не понятны
5. Эффективность



ТИПЫ VIEW



SIMPLE

- One table
- No functions
- No aggregation



COMPLEX



- Join tables
- Functions
- Aggregation

VIEW

```
CREATE OR REPLACE {FORCE | NOFORCE} VIEW  
schema.view_name (alias1, alias2, ...)  
AS subquery  
WITH CHECK OPTION {CONSTRAINT constrain_tname}  
WITH READ ONLY {CONSTRAINT constraint_name};
```

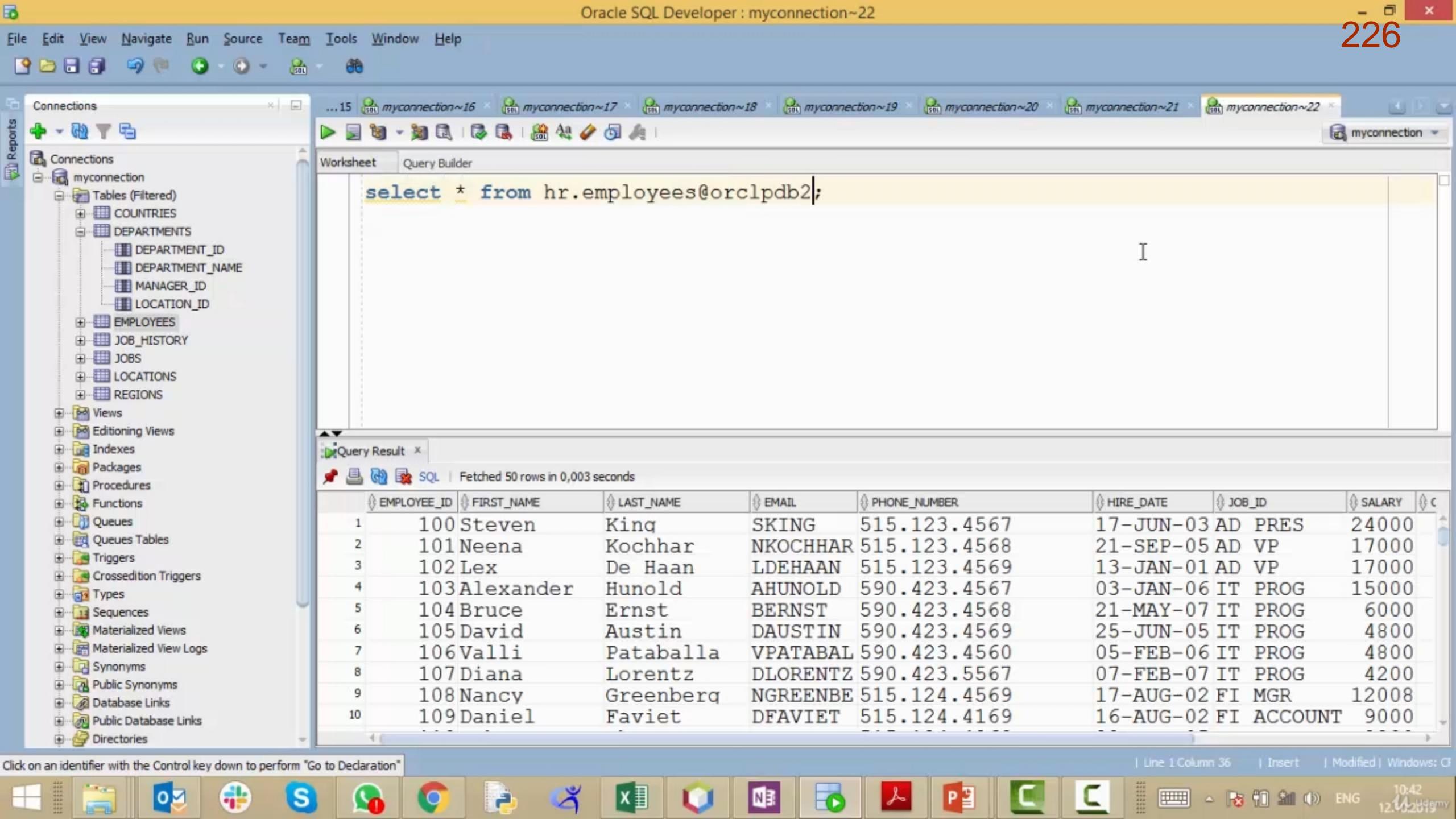
VIEW

```
CREATE OR REPLACE {FORCE | NOFORCE} VIEW  
schema.view_name (alias1, alias2, ...)  
AS subquery  
WITH CHECK OPTION {CONSTRAINT constraint_name}  
WITH READ ONLY {CONSTRAINT constraint_name};
```

```
ALTER VIEW schema.view_name COMPILE;
```

```
DROP VIEW schema.view_name;
```





SYNONYM

Записная книжка... > Заметки на полях

Virt ~~select * from q;~~
DML com.

↓ students ↓ ↓
id name course faculty id offices
1 2 3 4 5 6

Все
1) много схем
2) каких схем есть

~~Students~~ ↓
~~id name course faculty~~

primary key INT
unique
~~faculty~~ select

SYNONYM



Private

Schemaname Synname

Public

Synname

Select * from Syn,

- 1) Schemame
- 2) public syn

File Edit View Navigate Run Source Team Tools Window Help

C:\WINDOWS\system32\cmd.exe - sqlplus sys/test1@orclpdb2 as sysdba

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\zaurtregulov>sqlplus sys/test1@orclpdb2 as sysdba
SQL*Plus: Release 12.2.0.1.0 Production on Sat Oct 12 10:55:32 2019
Copyright (c) 1982, 2016, Oracle. All rights reserved.

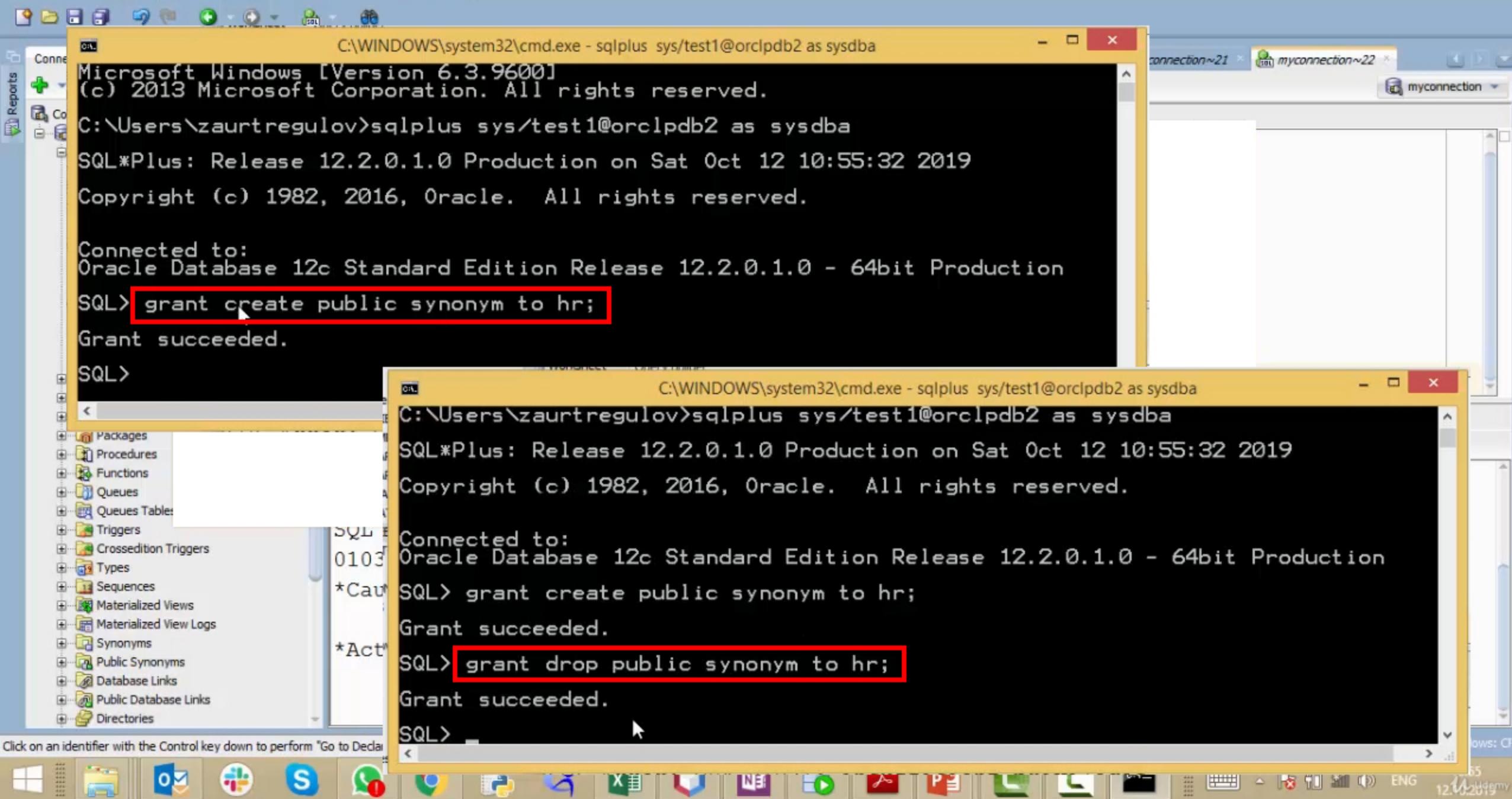
Connected to:
Oracle Database 12c Standard Edition Release 12.2.0.1.0 - 64bit Production
SQL> grant create public synonym to hr;
Grant succeeded.
SQL>
```

C:\WINDOWS\system32\cmd.exe - sqlplus sys/test1@orclpdb2 as sysdba

```
C:\Users\zaurtregulov>sqlplus sys/test1@orclpdb2 as sysdba
SQL*Plus: Release 12.2.0.1.0 Production on Sat Oct 12 10:55:32 2019
Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Standard Edition Release 12.2.0.1.0 - 64bit Production
SQL> grant create public synonym to hr;
Grant succeeded.
SQL> grant drop public synonym to hr;
Grant succeeded.
SQL>
```

Click on an identifier with the Control key down to perform "Go to Declaration".



SYNONYM

```
CREATE PUBLIC SYNONYM synonym_name  
FOR object_name;
```

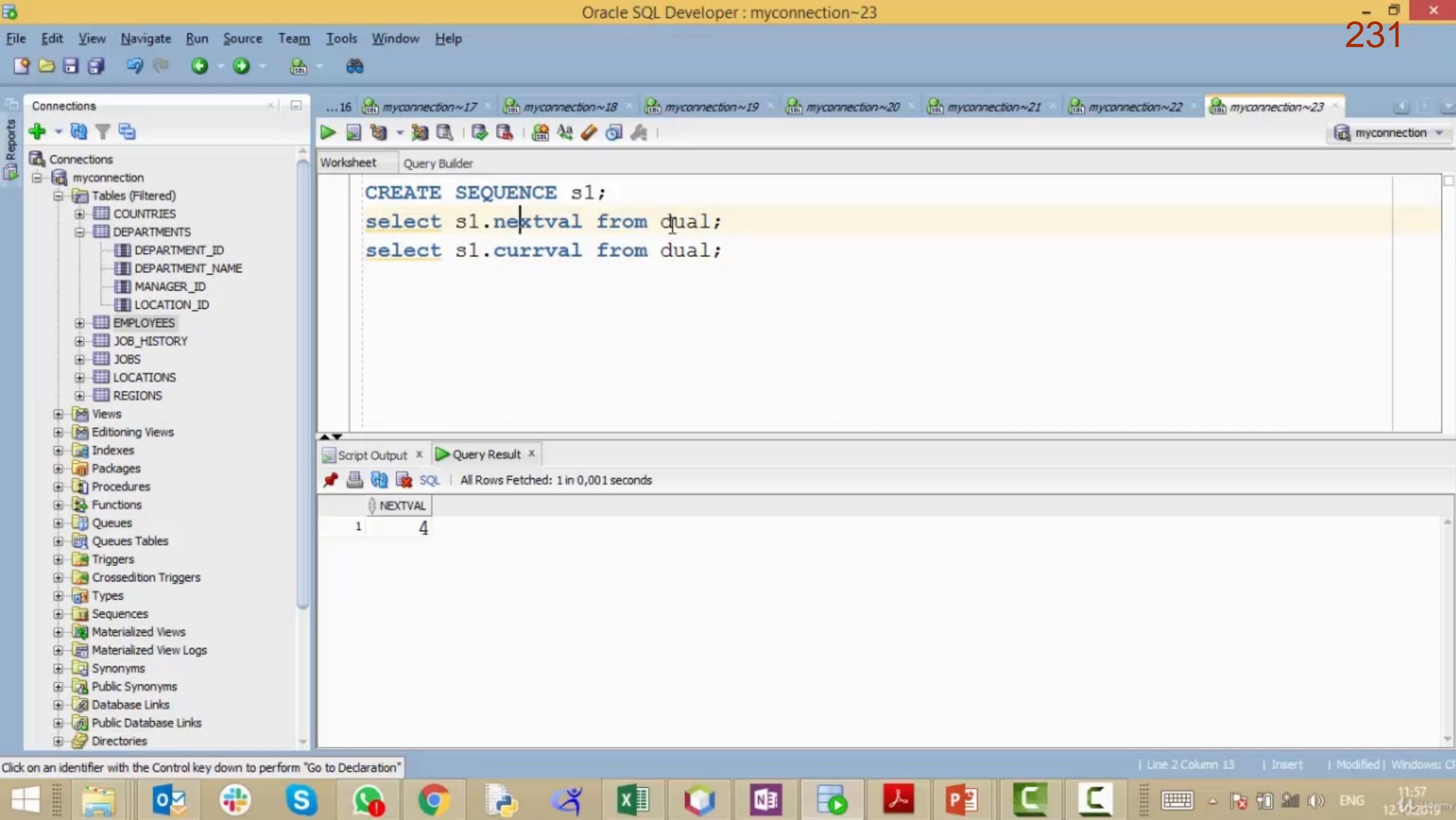
```
ALTER PUBLIC SYNONYM synonym_name  
COMPILE;
```

```
DROP PUBLIC SYNONYM synonym_name;
```



SEQUENCE

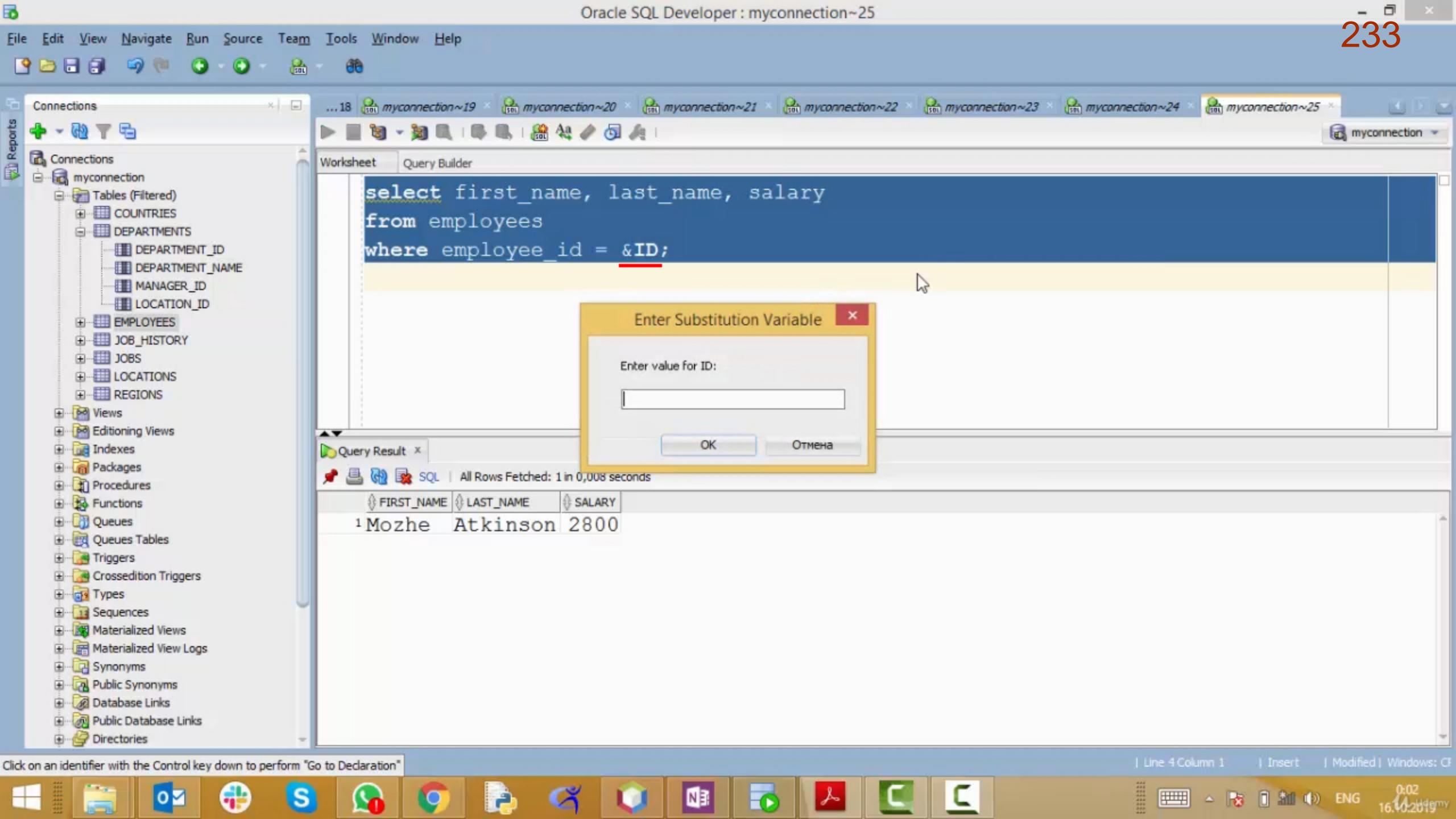
```
CREATE SEQUENCE schema.sequence_name  
INCREMENT BY number  
START WITH number  
{MAXVALUE number | NOMAXVALUE}  
{MINVALUE number | NOMINVALUE}  
{CYCLE | NOCYCLE}  
{CACHE number | NOCACHE};
```

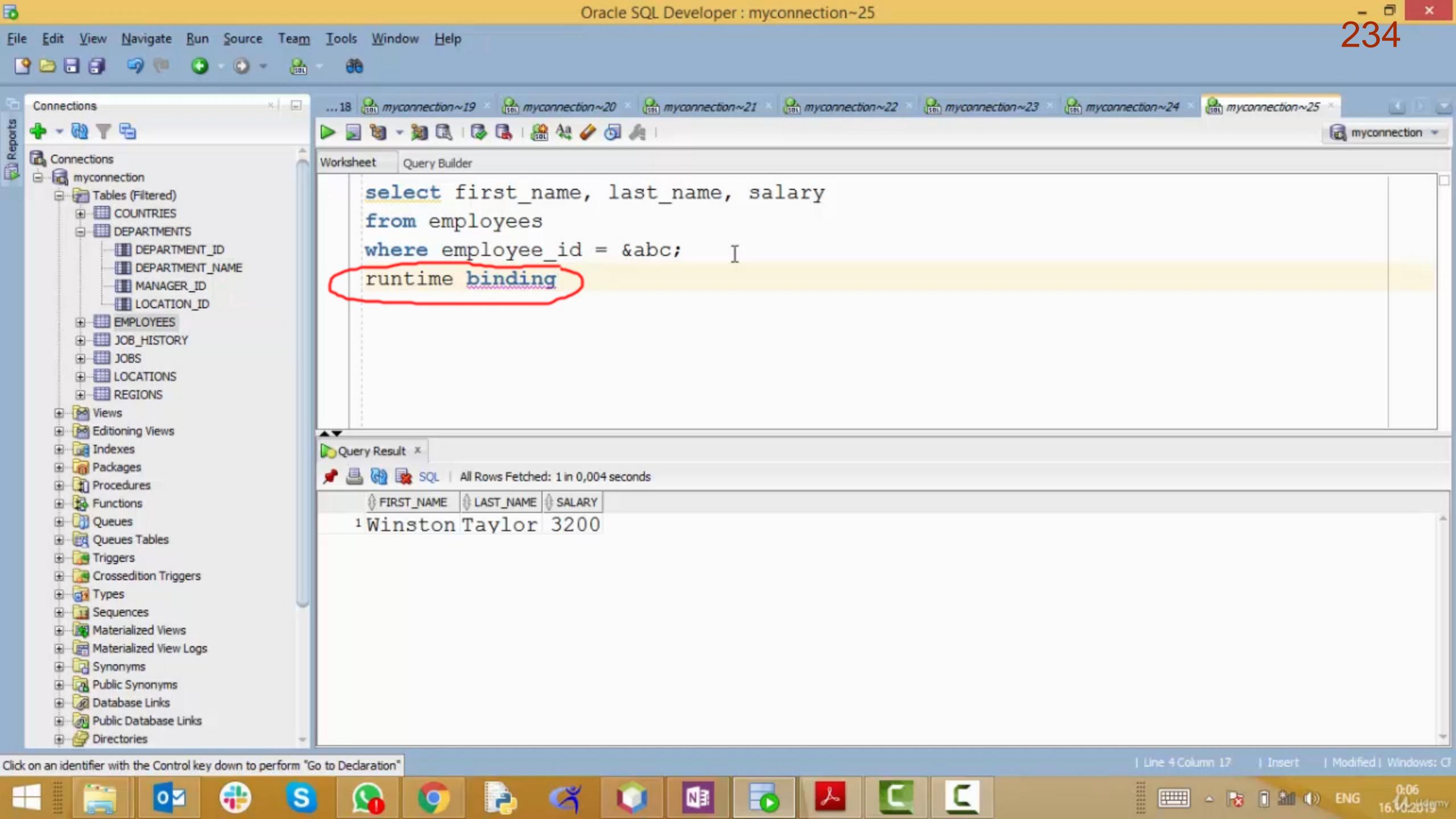


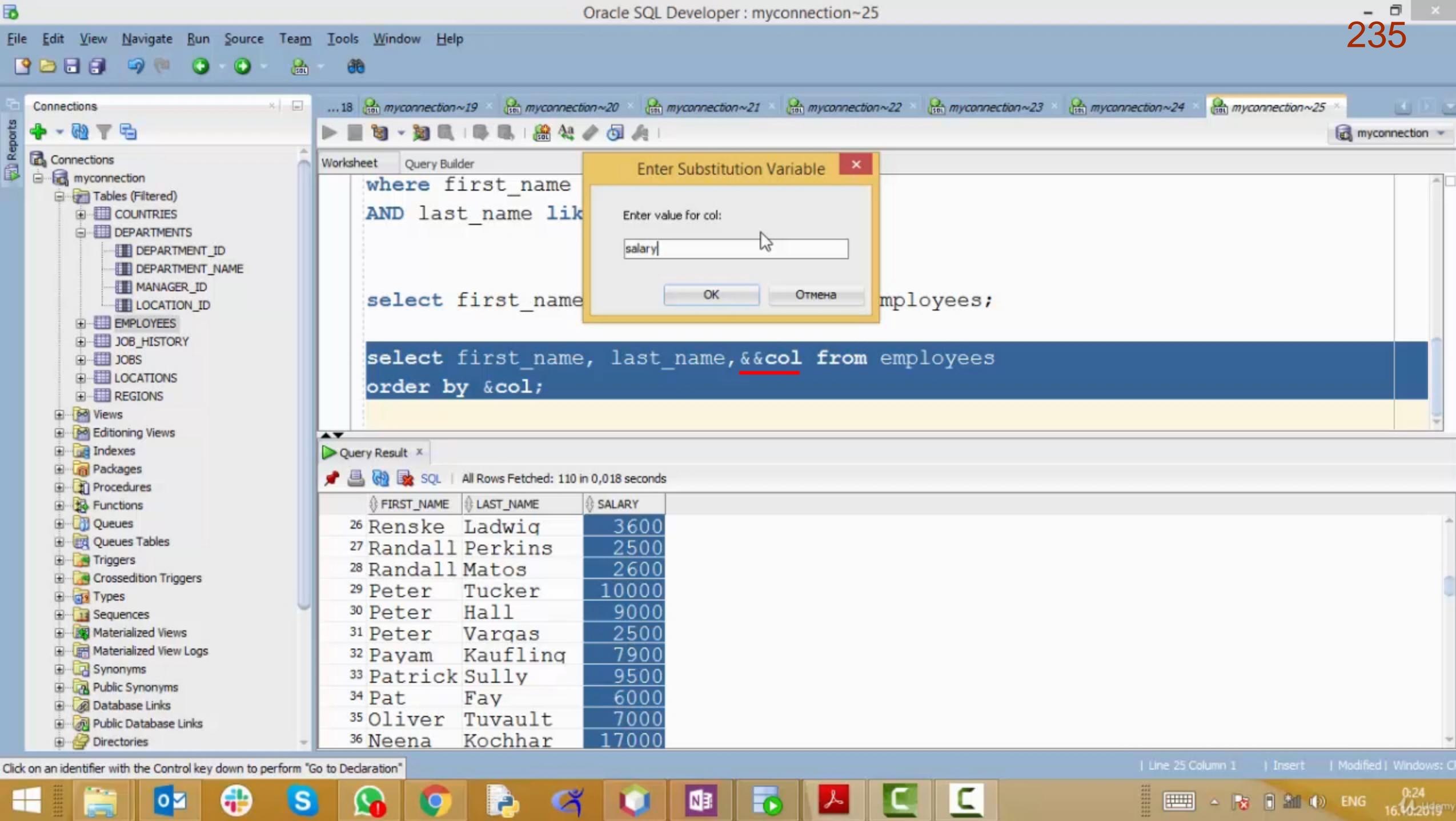
SEQUENCE

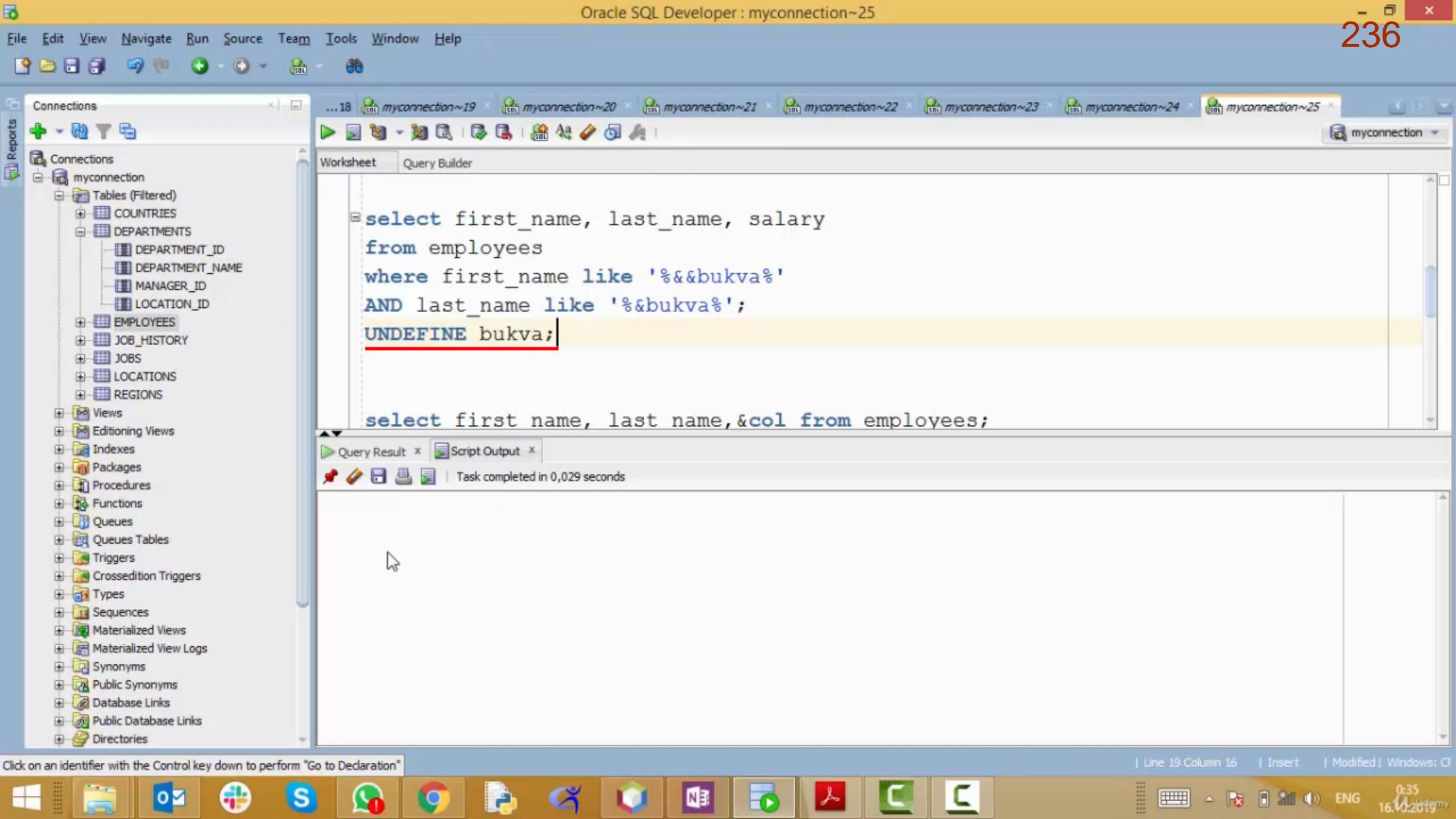
```
ALTER SEQUENCE schema.sequence_name  
INCREMENT BY number  
{MAXVALUE number | NOMAXVALUE}  
{MINVALUE number | NOMINVALUE}  
{CYCLE | NOCYCLE}  
{CACHE number | NOCACHE};
```

```
DROP SEQUENCE schema.sequence_name;
```









Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

```

select first_name, last_name, salary
from employees
where first_name like '%&&bukva%'
AND last_name like '%&&bukva%';
UNDEFINE bukva;
DEFINE;
```

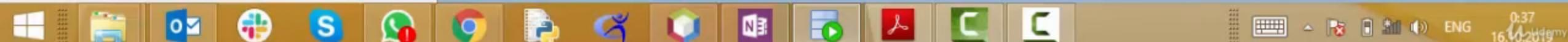
Script Output X | Query Result X
 Task completed in 0,021 seconds

```

DEFINE _EDITOR = "notepad" (CHAR)
DEFINE _O_VERSION = "Oracle Database 12c Standard Edition Release 12.2.0.1.0 - 64bit P
DEFINE _O_RELEASE = 1202000000 (NUMBER)
DEFINE _PWD = "D:\myoracle\product\12.2.0\dbhome_1\sqldeveloper\sqldeveloper\bin" (CHAR)
DEFINE COL = "salary" (CHAR)
DEFINE BUKVA = "s" (CHAR)
DEFINE BUKVA2 = "a" (CHAR)
```

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 20 Column 8 | Insert | Modified | Windows: C





Connections

+ Connections

myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```

select first_name, last_name, salary
from employees
where first_name like '%&&bukva%'
AND last_name like '%&&bukva%';
UNDEFINE bukva;
DEFINE;
DEFINE bukva = a; I
```

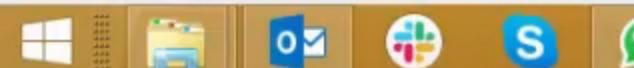
Script Output x | Task completed in 0,023 seconds

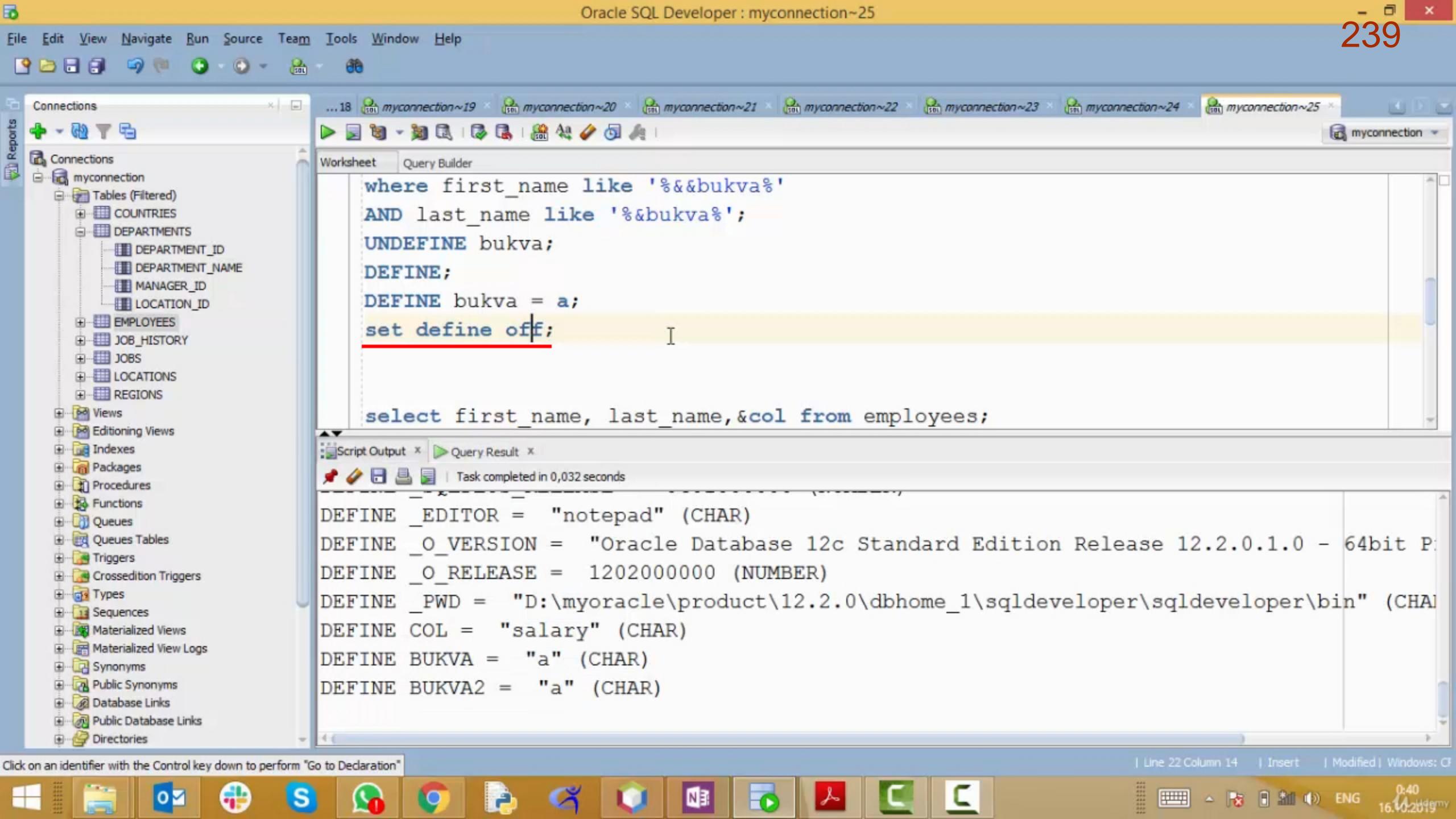
```

DEFINE _EDITOR = "notepad" (CHAR)
DEFINE _O_VERSION = "Oracle Database 12c Standard Edition Release 12.2.0.1.0 - 64bit P:
DEFINE _O_RELEASE = 1202000000 (NUMBER)
DEFINE _PWD = "D:\myoracle\product\12.2.0\dbhome_1\sqldeveloper\sqldeveloper\bin" (CHAR)
DEFINE COL = "salary" (CHAR)
DEFINE BUKVA = "t" (CHAR)
DEFINE BUKVA2 = "a" (CHAR)
```

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 21 Column 18 | Insert | Modified | Windows: C:





Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

```
Worksheet Query Builder
where first_name like '%&&bukva%'
AND last_name like '%&&bukva%';
UNDEFINE bukva;
DEFINE;
DEFINE bukva = a;
set define off;
set define on;
```

Script Output x | Task completed in 0,041 seconds

```
DEFINE _EDITOR = "notepad" (CHAR)
DEFINE _O_VERSION = "Oracle Database 12c Standard Edition Release 12.2.0.1.0 - 64bit P:
DEFINE _O_RELEASE = 1202000000 (NUMBER)
DEFINE _PWD = "D:\myoracle\product\12.2.0\dbhome_1\sqldeveloper\sqldeveloper\bin" (CHAR)
DEFINE COL = "salary" (CHAR)
DEFINE BUKVA = "a" (CHAR)
DEFINE BUKVA2 = "a" (CHAR)
```

File Edit View Navigate Run Source Team Tools Window Help

C:\WINDOWS\system32\cmd.exe - sqlplus hr/hr@orclpdb2

```
Michael          13000  
Shelley          12008
```

16 rows selected.

```
SQL> set verify off;  
SQL> select first_name, salary from employees where salary > &sal;  
Enter value for sal: 10000
```

FIRST_NAME	SALARY
Steven	24000
Neena	17000
Lex	17000
Alexander	15000
Nancy	12008
Den	11000
John	14000
Karen	13500

Indexes
Packages
Procedures
Functions
Queues
Queues Tables
Triggers
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories

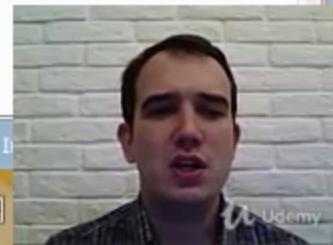
Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0,001 seconds

	FIRST_NAME	LAST_NAME	SALARY
1	Stephen	Stiles	3200
2	Nanette	Cambrault	7500
3	Elizabeth	Bates	7300
4	Timothy	Gates	2900
5	Britney	Everett	3900

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 24 Column 1 | I



File Edit View Navigate Run Source Team Tools Window Help

C:\WINDOWS\system32\cmd.exe - sqlplus hr/hr@orclpdb2

```
Ellen          11000
Michael        13000
Shelley        12008

16 rows selected.

SQL> set verify on;
SQL> select first_name, salary from employees where salary > &sal;
Enter value for sal: 15000
old  1: select first_name, salary from employees where salary > &sal
new  1: select first_name, salary from employees where salary > 15000
FIRST_NAME      SALARY
-----
Steven          24000
Neena           17000
Lex              17000

SQL>
```



Indexes
Packages
Procedures
Functions
Queues
Queues Tables
Triggers
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0,001 seconds

	FIRST_NAME	LAST_NAME	SALARY
1	Stephen	Stiles	3200
2	Nanette	Cambrault	7500
3	Elizabeth	Bates	7300
4	Timothy	Gates	2900
5	Britney	Everett	3900

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 24 Column 1 | I



File Edit View Navigate Run Source Team Tools Window Help



Connections



Connections

myconnection

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Views

Editoring Views

Indexes

Packages

Procedures

Functions

Queues

Queues Tables

Triggers

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Editions

Java

XML Schemas

XML DB Repository

hw_12.sql

SQL Worksheet History



myconnection

Worksheet Query Builder

```
) ;

UPDATE test200 SET &col = &val WHERE id = &id_val;
set verify off;
SELECT * FROM test200
WHERE name1 = '&&val1' AND name2 = '&val1'
AND address1 = '&&val2' AND address2 = '&val2';
```

Script Output

Task completed in 0,017 seconds

Table TEST200 created.

```
old:UPDATE test200 SET &col = &val WHERE id = &id_val
new:UPDATE test200 SET name1 = 'abc' WHERE id = 3
```

0 rows updated.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 10 Column 16 | Insert | Modified | Windows: C



1:09
20.10.2019

ENG

File Edit View Navigate Run Source Team Tools Window Help



Connections

- + Connections
 - myconnection
 - Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
 - Views
 - Editoring Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Queues
 - Queues Tables
 - Triggers
 - Crossedition Triggers
 - Types
 - Sequences
 - Materialized Views
 - Materialized View Logs
 - Synonyms
 - Public Synonyms
 - Database Links
 - Public Database Links
 - Directories
 - Editions
 - Java
 - XML Schemas
 - XML DB Repository

hw_12.sql

SQL Worksheet History



myconnection

Worksheet Query Builder

```
) ;  
  
UPDATE test200 SET &col = &val WHERE id = &id_val;  
set verify on;  
SELECT * FROM test200  
WHERE name1 = '&&val1' AND name2 = '&val1'  
AND address1 = '&&val2' AND address2 = '&val2';
```

Script Output

Task completed in 0,018 seconds

```
old:UPDATE test200 SET &col = &val WHERE id = &id_val  
new:UPDATE test200 SET name1 = 'abc' WHERE id = 3
```

0 rows updated.

0 rows updated.

Click on an identifier with the Control key down to perform "Go to Declaration"

| Line 10 Column 14 | Insert | Modified | Windows: C



1:10 ENG 20.10.2019

Connections

+ myconnection

- Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
- Views
- Editing Views
- Indexes
- Packages
- Procedures
- Functions
- Queues
- Queues Tables
- Triggers
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories

Worksheet Query Builder

```
UNDEFINE bukva;
DEFINE;
DEFINE bukva = a;
set define off;
set define on;

sql language commands
sql client control commands
```

Script Output X | Query Result X

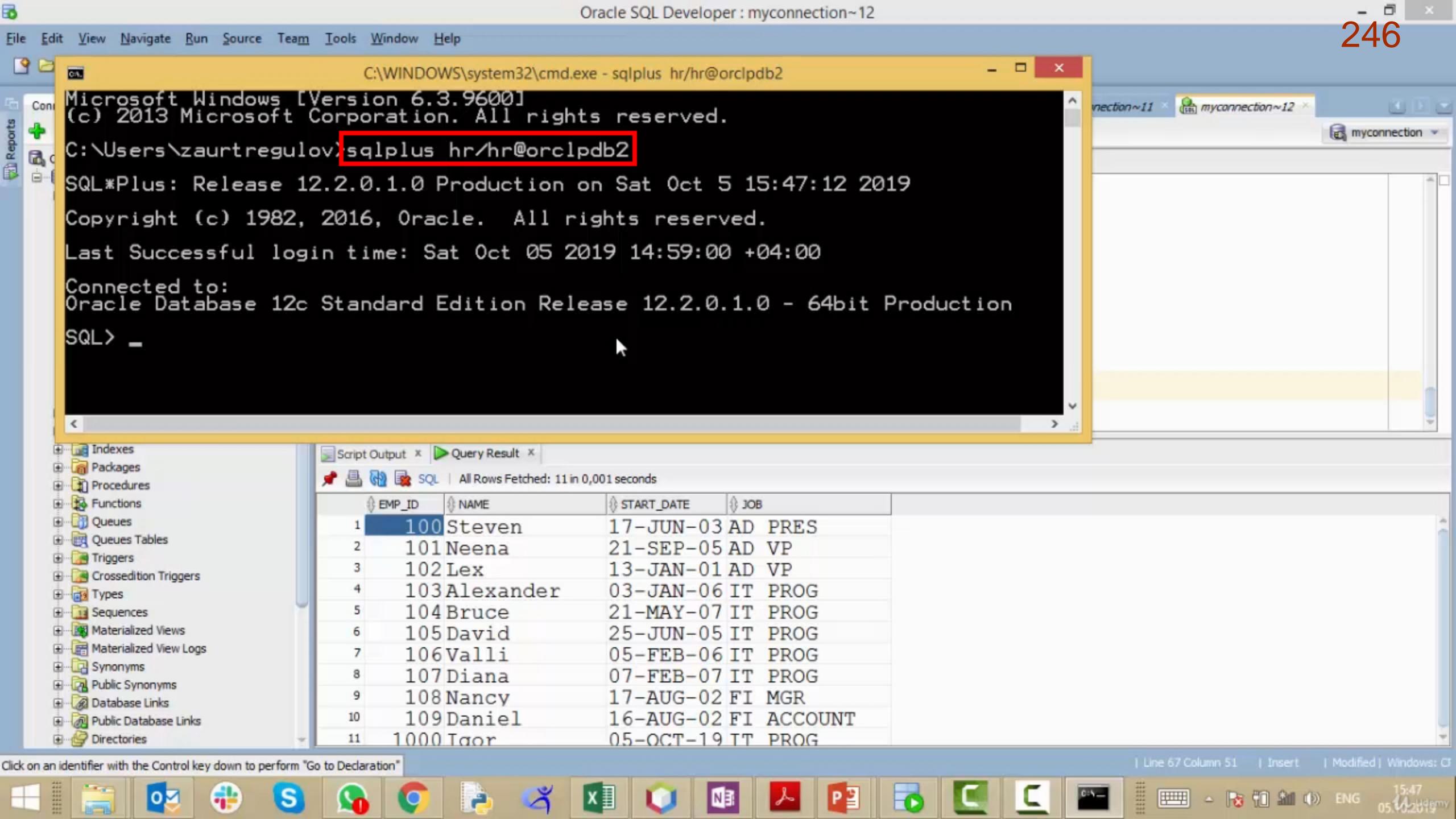
All Rows Fetched: 5 in 0,001 seconds

	FIRST_NAME	LAST_NAME	SALARY
1	Stephen	Stiles	3200
2	Nanette	Cambrault	7500
3	Elizabeth	Bates	7300
4	Timothy	Gates	2900
5	Britney	Everett	3900

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 26 Column 28 | D





File Edit View Navigate Run Source Team Tools Window Help



Connections

