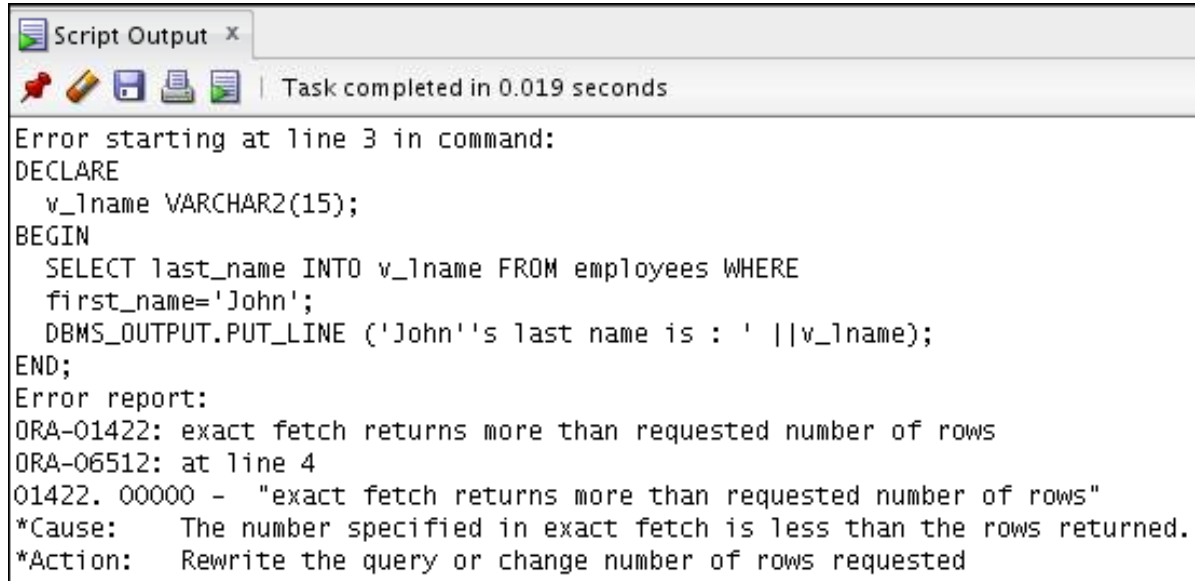


Что такое исключение?

```
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname
  FROM employees
  WHERE first_name='John';
  DBMS_OUTPUT.PUT_LINE ('John''s last name is : ' || v_lname);
END;
```



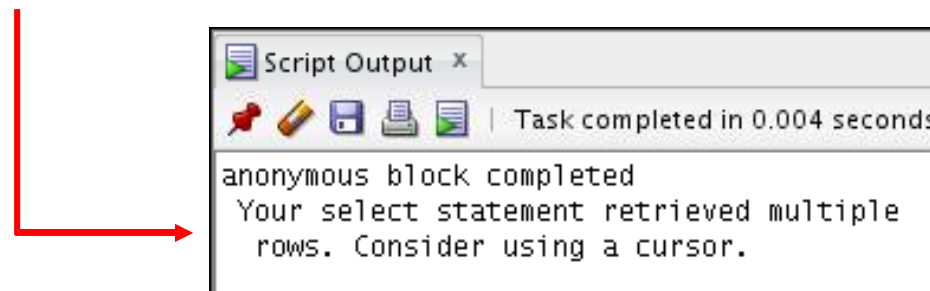
The screenshot shows a 'Script Output' window with a title bar and standard icons. Below the title bar, it states 'Task completed in 0.019 seconds'. The main text area displays the following error message:

```
Error starting at line 3 in command:
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname FROM employees WHERE
  first_name='John';
  DBMS_OUTPUT.PUT_LINE ('John''s last name is : ' || v_lname);
END;
Error report:
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at line 4
01422. 00000 - "exact fetch returns more than requested number of rows"
*Cause:      The number specified in exact fetch is less than the rows returned.
*Action:     Rewrite the query or change number of rows requested
```

A red L-shaped arrow points from the left margin towards the error message in the screenshot.

Обработка исключений: пример

```
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname
  FROM employees
  WHERE first_name='John';
  DBMS_OUTPUT.PUT_LINE ('John''s last name is : ' || v_lname);
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE (' Your select statement retrieved
    multiple rows. Consider using a cursor. ');
END;
/
```



Исключения в PL/SQL

- Исключение (exception) – ошибка, которая возникает в процессе выполнения программы.
- Исключение может быть вызвано:
 - Сервером Oracle
 - Напрямую в программе
- Исключение можно обработать:
 - «Поймать» обработчиком
 - Или передать в вызывающее окружение

Типы исключений

- Предопределенные Oracle
- Не определенные Oracle
- Пользовательские



Вызываются по
ошибке

Вызываются явно

Обработка исключений

EXCEPTION

```
WHEN exception1 [OR exception2 . . .] THEN
    statement1;
    statement2;
    . . .
[WHEN exception3 [OR exception4 . . .] THEN
    statement1;
    statement2;
    . . .]
[WHEN OTHERS THEN
    statement1;
    statement2;
    . . .]
```

Рекомендации по обработке исключений

- Секция обработки исключений начинается с ключевого слова `EXCEPTION`
- Можно добавлять несколько обработчиков
- Срабатывает только один обработчик перед выходом из секции обработки
- `WHEN OTHERS` используйте последним

Обработка predefined исключений

- Указать название исключения WHEN `exception_name` в секции обработки
- Пример predefined исключений:
 - `NO_DATA_FOUND`
 - `TOO_MANY_ROWS`
 - `INVALID_CURSOR`
 - `ZERO_DIVIDE`
 - `DUP_VAL_ON_INDEX`

Функции для обработки исключений

- `SQLCODE`: возвращает код ошибки
- `SQLERRM`: возвращает текст ошибки

Функции для обработки исключений

```
DECLARE
    error_code          NUMBER;
    error_message       VARCHAR2 (255);
BEGIN
    ...
EXCEPTION
    ...
    WHEN OTHERS THEN
        ROLLBACK;
        error_code := SQLCODE ;
        error_message := SQLERRM ;
        INSERT INTO errors (e_user, e_date, error_code,
            error_message) VALUES (USER, SYSDATE, error_code,
            error_message);
END;
/
```