upGrad

#LifeKoKaroLift

TechM Full Stack Software Development

upGrad

Course: Foundation of Database
Lecture On: Basic PL/SQL
Instructor: Vishwa Mohan

upGrad

Today's Agenda

1. **Introduction to PL/SQL**

   What and Why

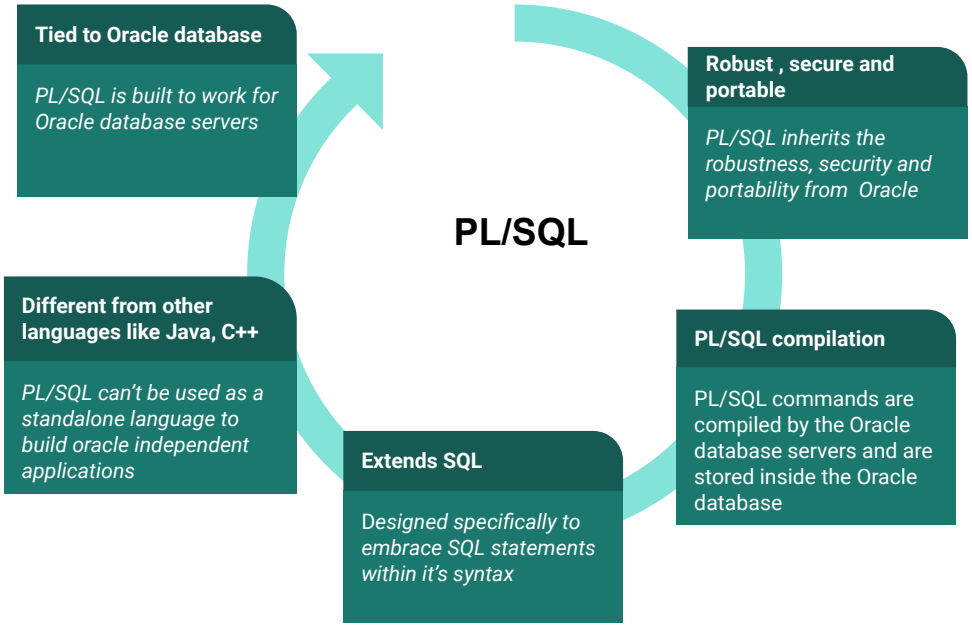
2. **Data types in PL/SQL**
3. **Conditionals in PL/SQL**
4. **PL/SQL blocks**

ORACLE DATABASE

PL/SQL

What is PL/SQL ?

It stands for **Procedural language** for **SQL**



vs

1. It is a database programming language.
2. It is application oriented.
3. It supports programming features such as control/loop.
4. PL/SQL cannot be embedded in SQL.

1. It is a database structured query language.
2. It is data-oriented.
3. It does not support the control structures.
4. SQL can be embedded in PL/SQL.

PL/SQL vs Other Programming Languages

upGrad

1. PL/SQL can be used only with Oracle database servers

2. PL/SQL program is compiled by the oracle database servers itself

3. It's an embedded language, built only to develop highly optimized/performing applications highly coupled with Oracle

1. They can be used with different databases like MySql, mongo, including Oracle db

2. These have their own dedicated runtime environment. For example, Java needs JRE for it's execution

3. They are very generic languages, used to build variety of standalone and web applications.

Advantages Of PL/SQL

1. It is tightly integrated with SQL.
2. It is a portable language for Oracle Database Development.
3. It is a high-performance language.
4. It makes the understanding and development of Oracle-based applications easy.
5. It enables better manageability.
6. It provides support for object-oriented programming.

Poll 1

Can PL/SQL code be run without Oracle Server environment?

1. Yes

2. No

Poll 1 (Answer)

Can PL/SQL code be run without Oracle Server environment?

1. Yes

2. No

Data Types

upGrad

Following are the types of data:

1. Scalar
   Scalars are of the following types:

   Numeric
   Boolean
   Character & String
   Date & Time

2. Composites
   Composites are of the following types:

   Record
   Collection

3. Large object (LOB)

   These are pointers to the large objects that are stored separately.

4. Reference

   This is a pointer to other data items.

Data Types: Scalar

upGrad

Numeric data type

Numeric data type represents numbers, integers and floating-point numbers.

This type of data is stored as NUMBER (*sql type*) and PLS_INTEGER (*specific to PL/SQL*).

PLS_INTEGER is faster than NUMBER because it uses hardware arithmetic.

PLS_INteGER data type is more space-optimised.

Data Types: Scalar

upGrad

The following are the subtypes of numeric type:

1. **NATURAL:** Non-negative PLS_INTEGER values
2. **NATURALN:** Non-negative PLS_INTEGER values with a NOT NULL constraint
3. **POSITIVE:** Positive PLS_INTEGER values
4. **POSITIVEN:** Positive PLS_INTEGER value with a NOT NULL constraint
5. **SIGNTYPE:** Three values -1, 0 or 1 that are useful for tri-state logic programming
6. **SIMPLE_INTEGER:** PLS_INTEGER values with a NOT NULL constraint

Data Types: Scalar

upGrad

BOOLEAN data type

This data type has the following three possible values :

1. TRUE
2. FALSE
3. NULL

Data Types: Scalar

upGrad

Character data type

This type has the following two data types:

1. CHAR(n)
2. VARCHAR2(n)

Data Types: Scalar

upGrad

DATETIME data type:

This type has the following six data types:

1. DATE (Ex : '1998-12-25')
2. TIMESTAMP (Ex : '1997-01-31 09:26:50.12')
3. TIMESTAMP WITH TIME ZONE ('1999-01-15 8:00:00 -8:00')
4. TIMESTAMP WITH LOCAL TIME ZONE (2017-03-15 19:02:00)

Detailed reference : https://docs.oracle.com/cd/E11882_01/server.112/e10729/ch4datetime.htm#NLSPG238

Data Types: Scalar

upGrad

The data type synonyms are presented in the following table:

Data Type	Synonyms
NUMBER	DEC, DECIMAL, DOUBLE PRECISION, FLOAT, INTEGER, INT, NUMERIC, REAL, SMALLINT
CHAR	CHARACTER, STRING
VARCHAR2	VARCHAR

Poll 2

Scalar data type in PL/SQL:

1. Holds multiple values

2. Holds single value

3. Both 1 and 2

4. None of the above

Poll 2 (Answer)

Scalar data type in PL/SQL:

1. Holds multiple values

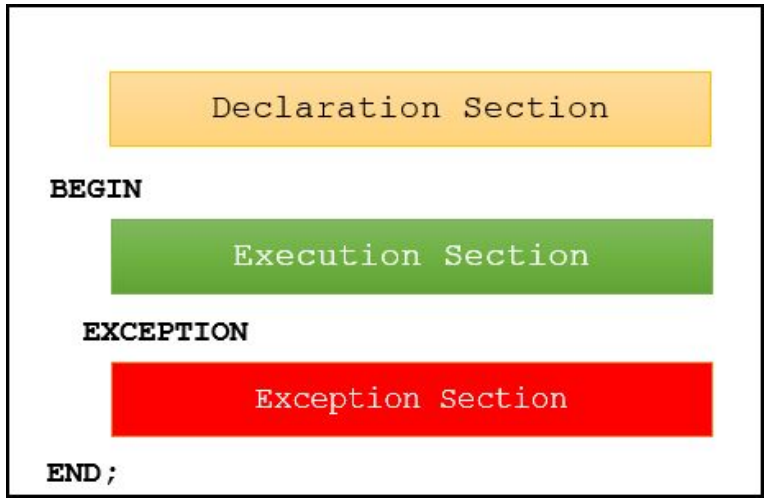
2. Holds single value

3. Both 1 and 2

4. None of the above

Blocks

upGrad



PL/SQL is a block-structured language where statements are organised in the form of blocks.

The following are the two types of blocks:

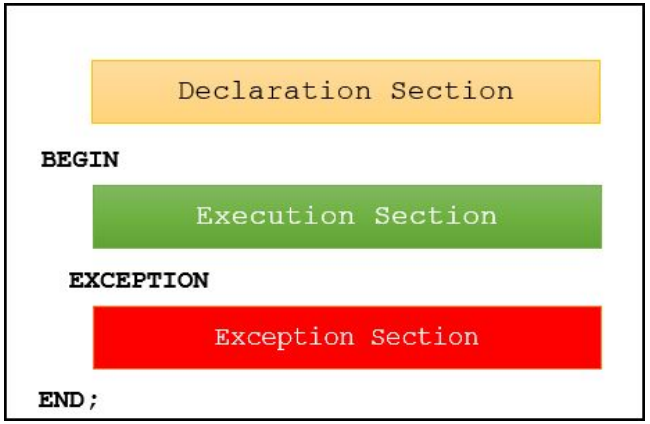
1. Named block
2. Anonymous block

A code block can have the following three sections:

1. Declaration (optional)
2. Execution (mandatory)
3. Exception (optional)

Blocks

upGrad



Named blocks

These blocks have mandatory headers or labels in the declaration section.

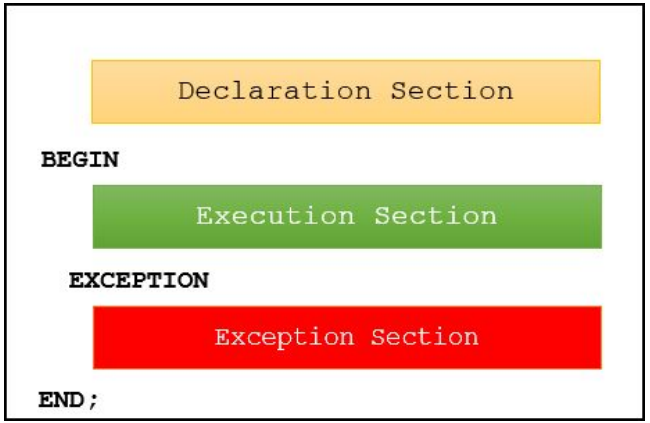
They can either be subprograms, such as functions, packages, or triggers*.

They are stored in the Oracle database server and can be used later.

*Are you intrigued by functions, packages and trigger? We will discuss them in the next session.

Blocks

upGrad



Anonymous blocks

These blocks do not have headers in the declaration section.

They do not form the body of a function, trigger or any procedure.

They are for one-time use and are not stored/saved in the Oracle Database Server.

Anonymous blocks are mostly used during debugging.

Let's take a look at Example 1.

Poll 3

Which of the below is true about any PL/SQL block?

1. It starts with declaration section which is mandatory.

2. It starts with declaration section which is optional.

3. Execution section is optional.

4. None of the above

Poll 3 (Answer)

Which of the below is true about any PL/SQL block?

1. It starts with declaration section which is mandatory.

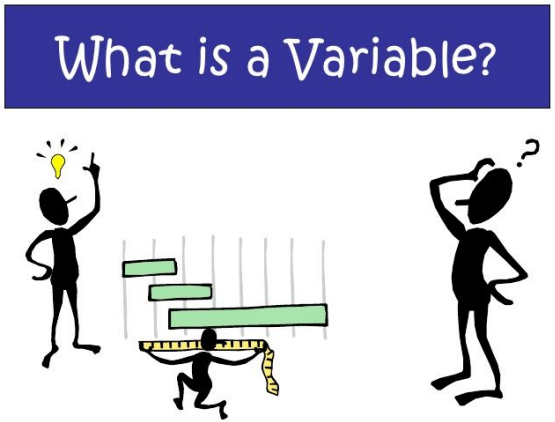
2. It starts with declaration section which is optional.

3. Execution section is optional.

4. None of the above

Variables

upGrad



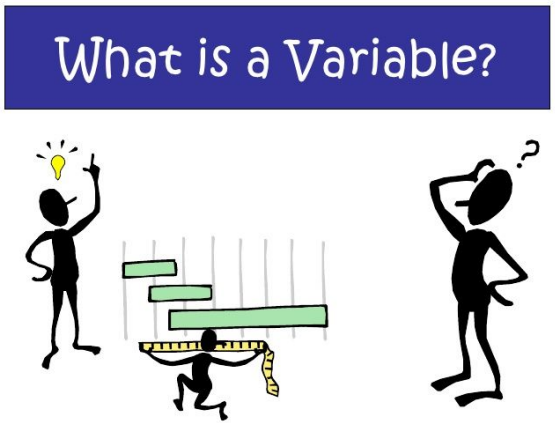
- These are named ‘storage locations’ and store values of a particular type.
- Before using a variable, it has to be declared in the declaration section.

The syntax used is as follows:

variable_name datatype [NOT NULL] [:= expression];

Notes:
The local variable name should start with **l_** and that of the global variable should start with **g_**.

Let’s take a look at Example 2.



Constant variable

Its value cannot change throughout the course of execution.

The syntax used is as follows:

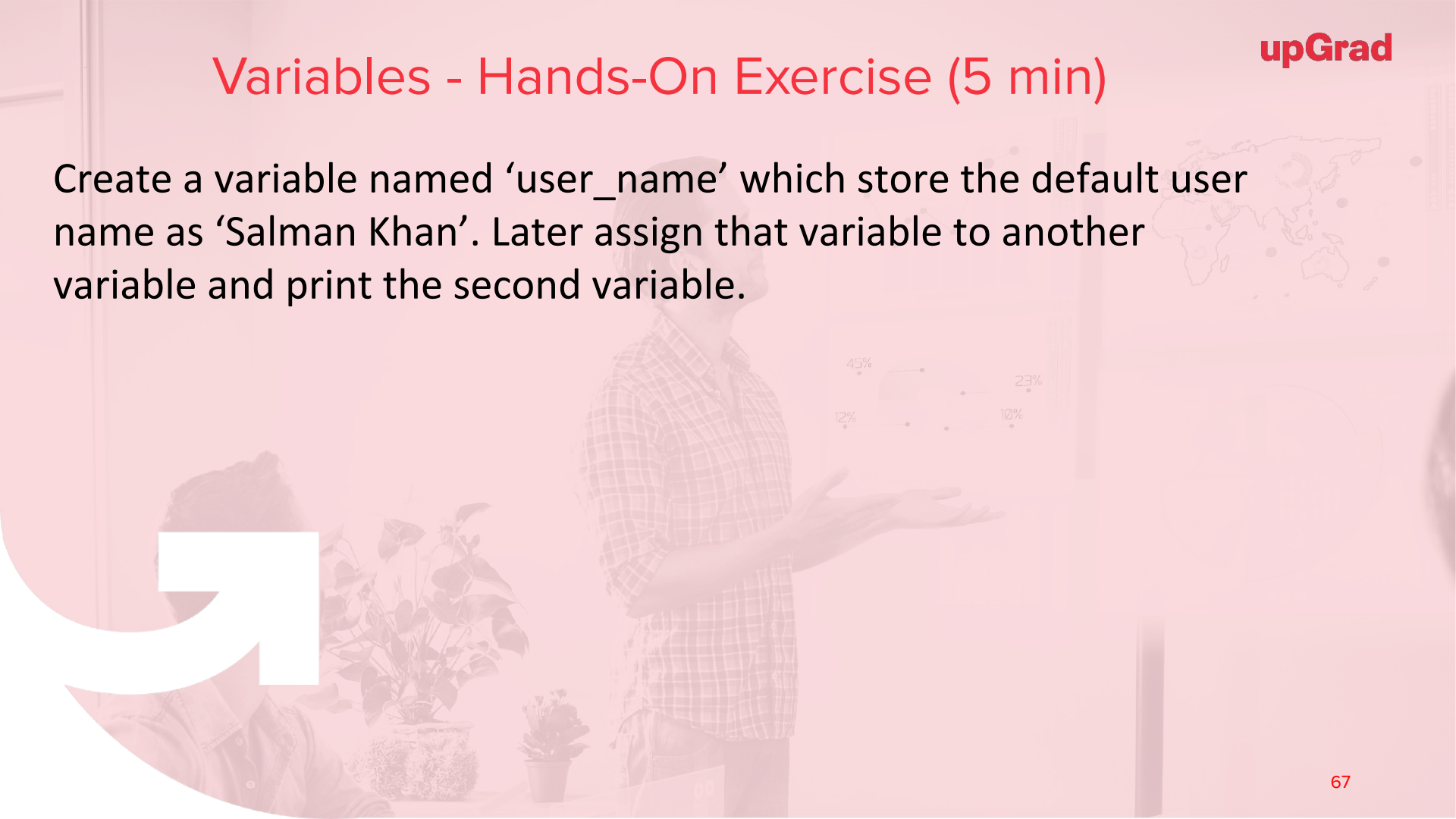
constant_variable CONSTANT datatype [NOT NULL] := expression

Let’s take a look at Example 3.

Variables - Hands-On Exercise (5 min)

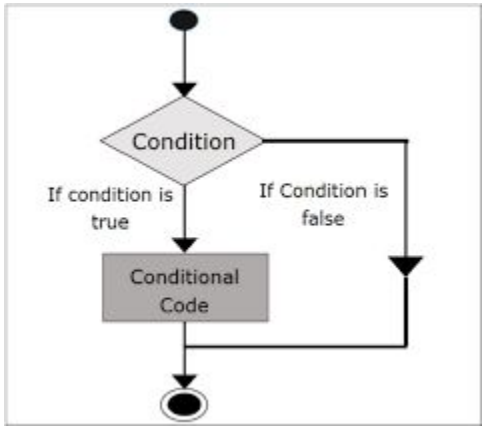
upGrad

Create a variable named ‘user_name’ which store the default user name as ‘Salman Khan’. Later assign that variable to another variable and print the second variable.



Conditionals

upGrad



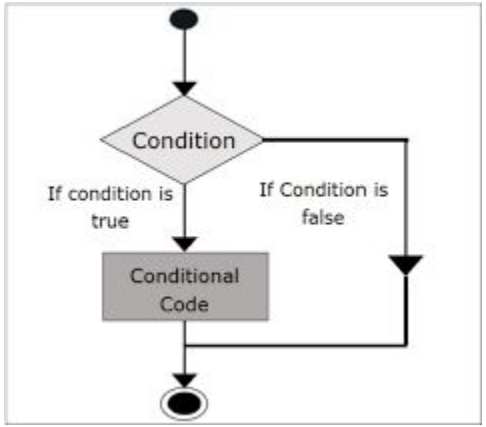
IF statements

These are used to decide whether to allow or skip a sequence of statements.

They have the following three forms:

1. IF THEN
2. IF THEN ELSE
3. IF THEN ELSIF

Conditionals



The syntax to be used for IF statements are as follows:

1. IF THEN

```
IF condition THEN
   statements;
END IF;
```

2. IF THEN ELSE

```
IF condition THEN
   statements;
ELSE
   else_statements;
END IF;
```

3. IF THEN ELSIF

```
IF condition_1 THEN
  statements_1
ELSIF condition_2 THEN
  statements_2
[ ELSE
    else_statements
]
END IF;
```

Let's take a look at example 4.

IF - Hands-On Exercise (5 min)

upGrad

Take a number and perform the following operations on it:

1. Check if it is divisible by 2, if yes, display output as ‘Number is divisible by 2’
2. Check if it is divisible by 3, if yes, display output as ‘Number is divisible by 3’
3. If it is divisible by neither 2 or 3, display output as ‘Number is neither divisible by 2 nor 3’

Poll 4

Which of the following is not a valid conditional keyword in PL/SQL?

1. IF
2. ELSEIF
3. ELSE
4. None of the above

Poll 4 (Answer)

Which of the following is not a valid conditional keyword in PL/SQL

1. IF
2. **ELSEIF**
3. ELSE
4. None of the above

Conditionals

upGrad

Case statement

It is used to decide on one of the ‘n’ different possible paths.

These statements are of the following two types:

1. Simple case statement
2. Searched case statement

Simple Case Statement

The simple case statement syntax is as follows:

```
CASE selector
WHEN selector_value_1 THEN
   statements_1
WHEN selector_value_1 THEN
   statement_2
ELSE
   else_statements
END CASE;
```

Let's take a look at Example 5.

Searched Case statement

The searched case statement syntax is as follows:

```
CASE
WHEN condition_1 THEN statements_1
WHEN condition_2 THEN statements_2
WHEN condition_n THEN statements_n
[ ELSE
  else_statements
]
END CASE;
```

Continuing Example 5

GOTO statement

This helps to transfer* control to the labelled block or statement.

The syntax used is as follows:

GOTO label_name;

*The GOTO statement cannot be used to transfer control to IF, CASE or LOOP statements.

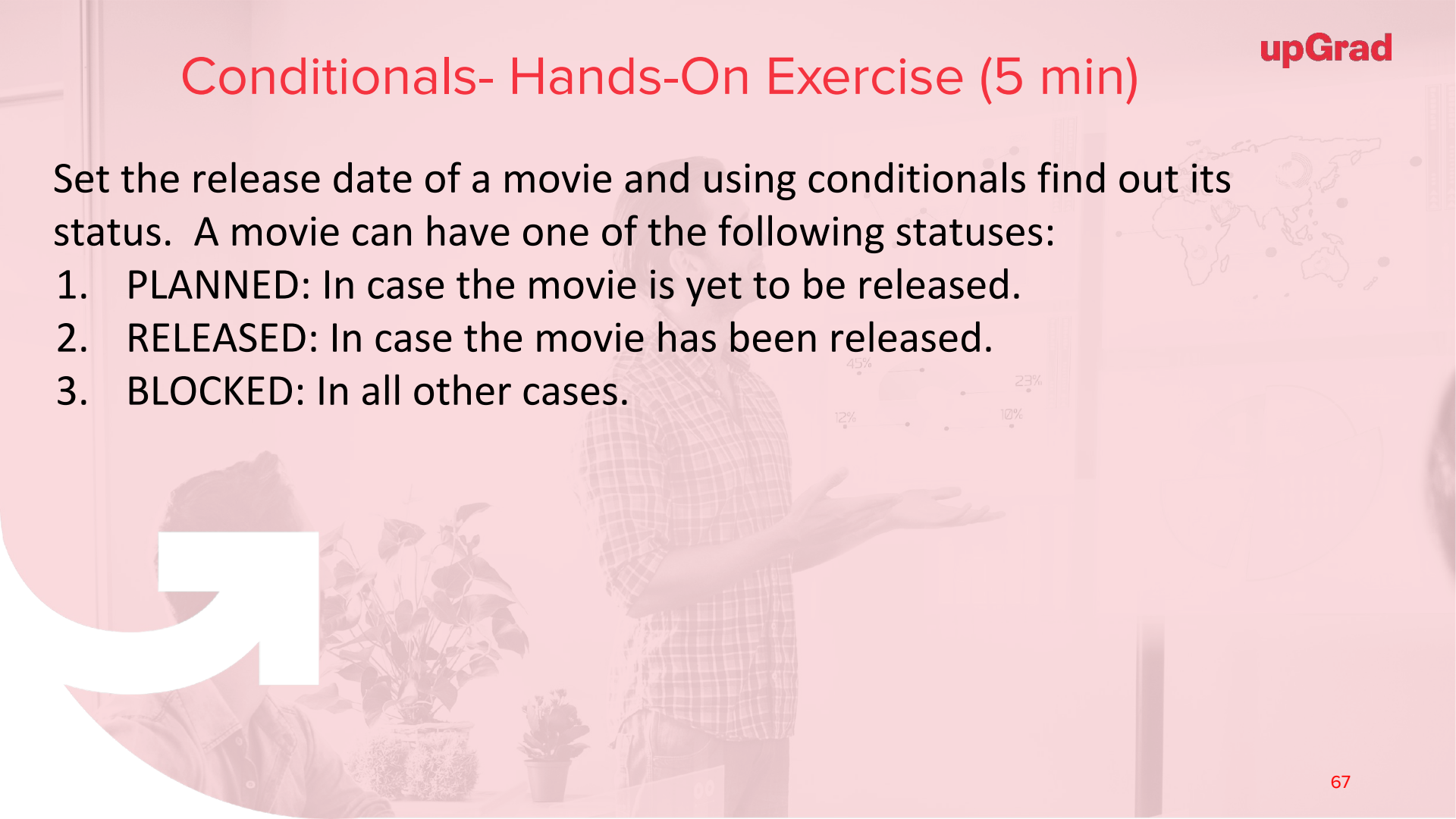
Let's take a look at example 6.

Conditionals- Hands-On Exercise (5 min)

upGrad

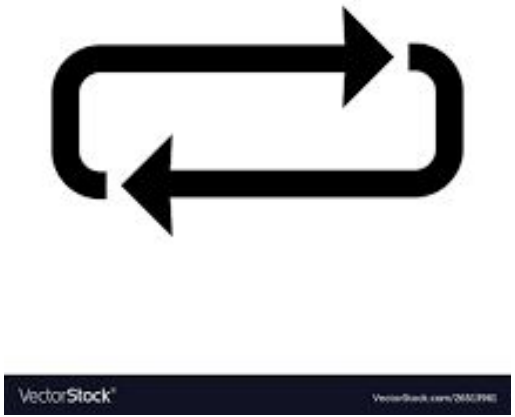
Set the release date of a movie and using conditionals find out its status. A movie can have one of the following statuses:

1. PLANNED: In case the movie is yet to be released.
2. RELEASED: In case the movie has been released.
3. BLOCKED: In all other cases.



LOOP Statement

upGrad



The LOOP statement involves repeating a block of code until the condition is satisfied

The LOOP statement

The syntax used is as follows:

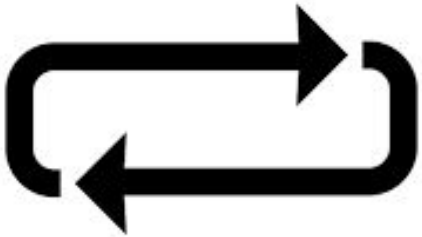
```
[label] LOOP
    statements;
END LOOP [loop_label];
```

*** The **EXIT** statement can be used to terminate the loop at any time.

When to use the LOOP statement:

1. When the body of a loop is to be executed at least once

Let's take a look at example 7.



The FOR LOOP statement involves repeating a block of code until the condition is satisfied.

The FOR LOOP statement

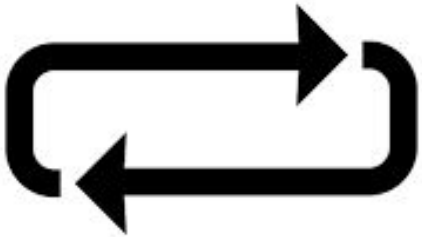
The syntax used is as follows:

```
FOR index IN start .. end LOOP
   statements;
END LOOP;
```

Notes:

1. The index is an local variable and cannot be used outside FOR LOOP.
2. Both start and end in a FOR LOOP statement are integers.

Let's take a look at example 8.



The WHILE LOOP statement involves repeating a block of code until the condition is satisfied.

The WHILE LOOP statement

The syntax used is as follows:

```
WHILE condition LOOP
   statements;
END LOOP;
```

Notes:

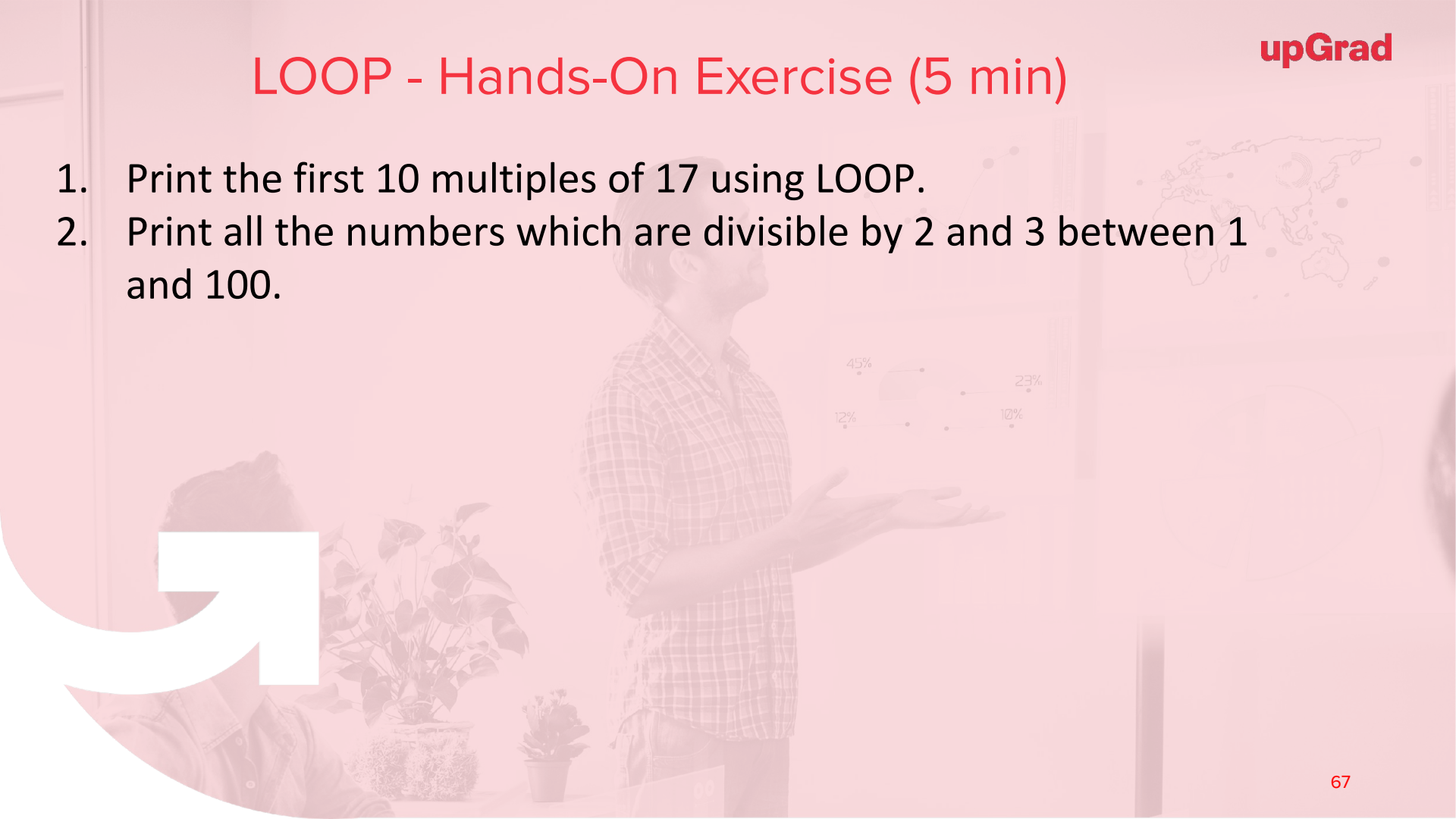
1. This will execute until the condition is TRUE and neither FALSE nor NULL.
2. **The CONTINUE statement** helps to exit the current loop iteration.

Let's take a look at example 9.

LOOP - Hands-On Exercise (5 min)

upGrad

1. Print the first 10 multiples of 17 using LOOP.
2. Print all the numbers which are divisible by 2 and 3 between 1 and 100.



Poll 5

GOTO statement can be used to transfer control inside for loop from outside the loop construct

1. True
2. False

Poll 5 (Answer)

GOTO statement can be used to transfer control inside for loop from outside the loop construct

1. True
2. **False**

SELECT INTO Statement

upGrad

SELECT INTO statement

This is used to fetch data from a single row of a table into the variables.

The syntax used is as follows:

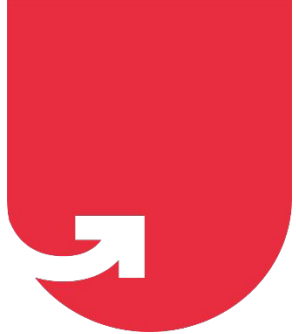
```
SELECT
  select_list
INTO
  variable_list
FROM
  table_name
WHERE
  condition;
```

Let's take a look at example 10.


upGrad
DISCUSSION TIME

upGrad

#LifeKoKaroLift



Thank You!