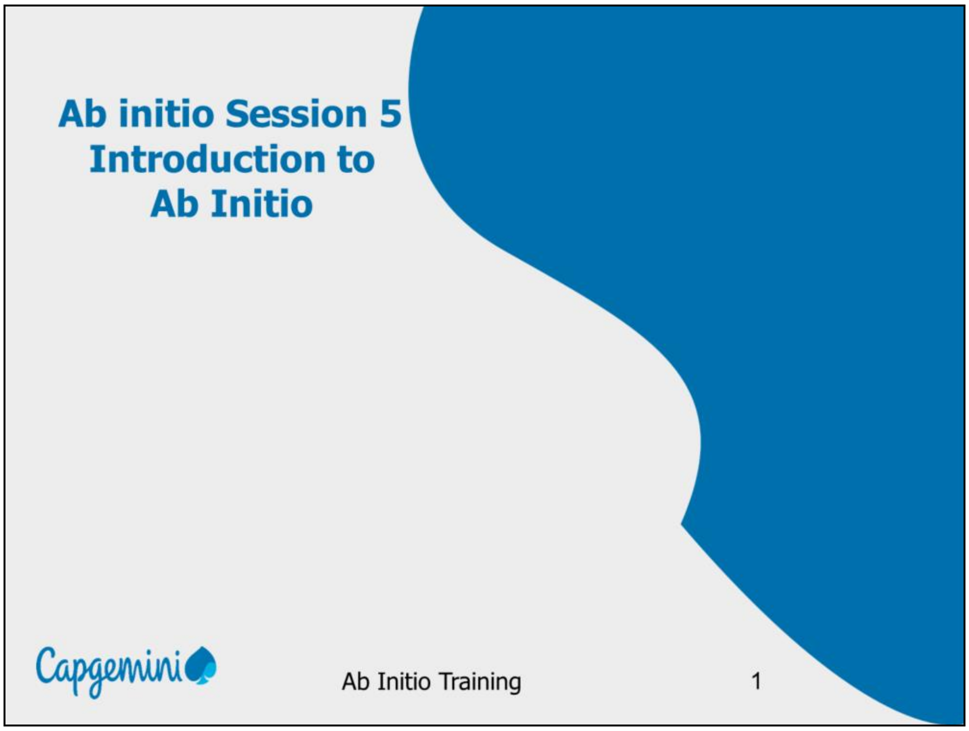
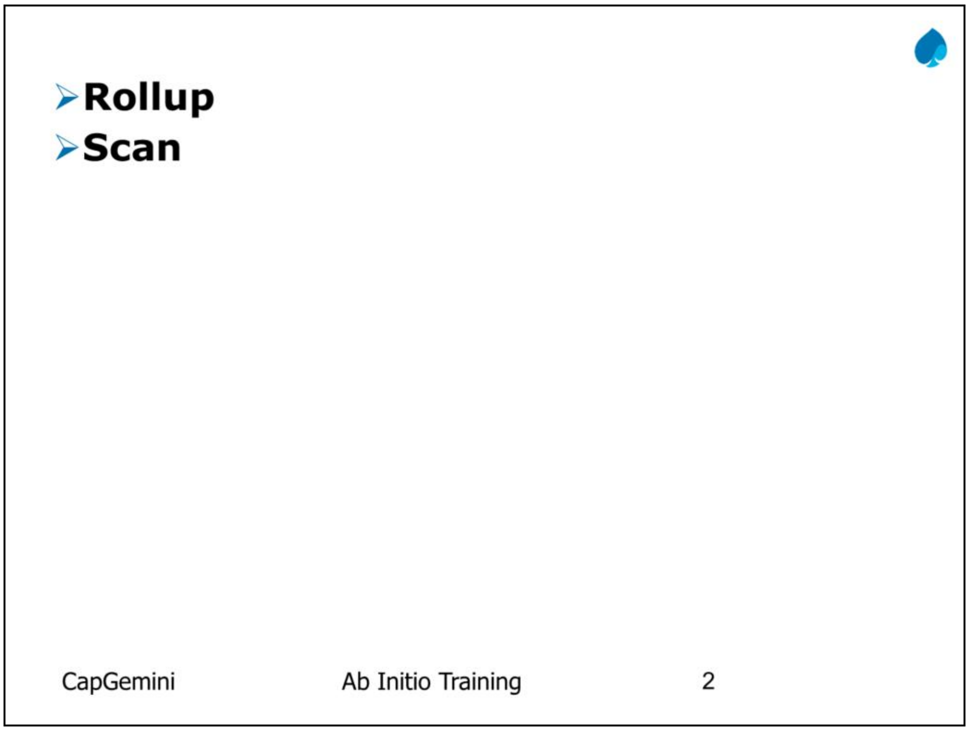


Ab initio Session 5 Introduction to Ab Initio

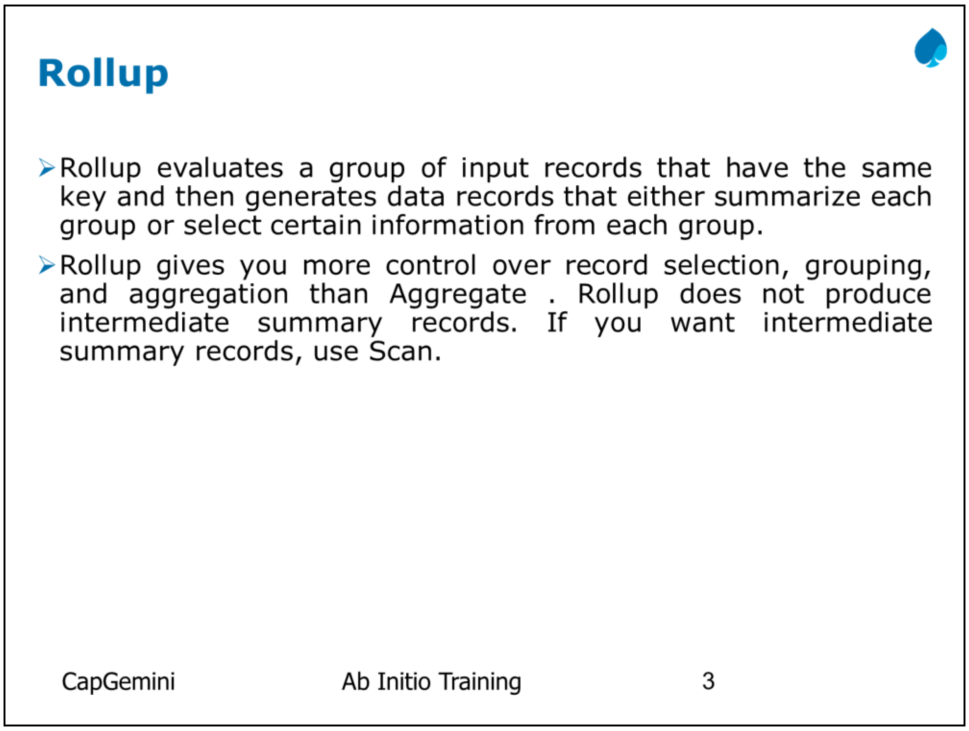
Capgemini

Ab Initio Training 1



- **Rollup**
- **Scan**

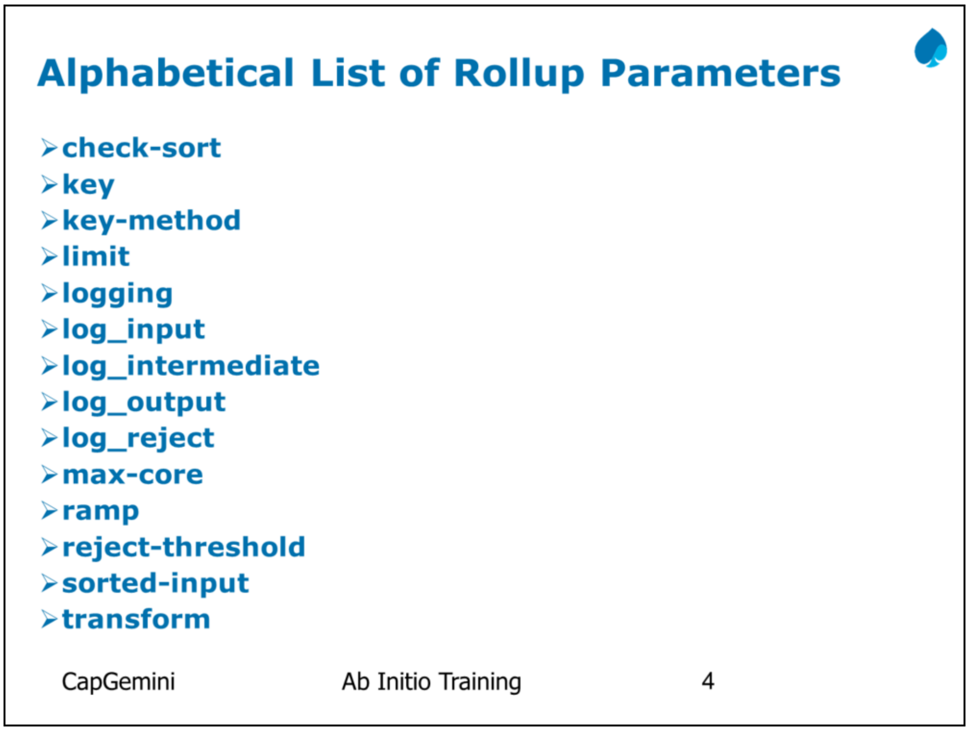
CapGemini Ab Initio Training 2



Rollup

- Rollup evaluates a group of input records that have the same key and then generates data records that either summarize each group or select certain information from each group.
- Rollup gives you more control over record selection, grouping, and aggregation than Aggregate . Rollup does not produce intermediate summary records. If you want intermediate summary records, use Scan.

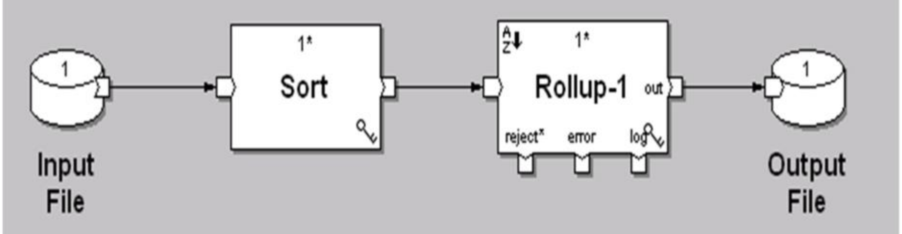
CapGemini Ab Initio Training 3

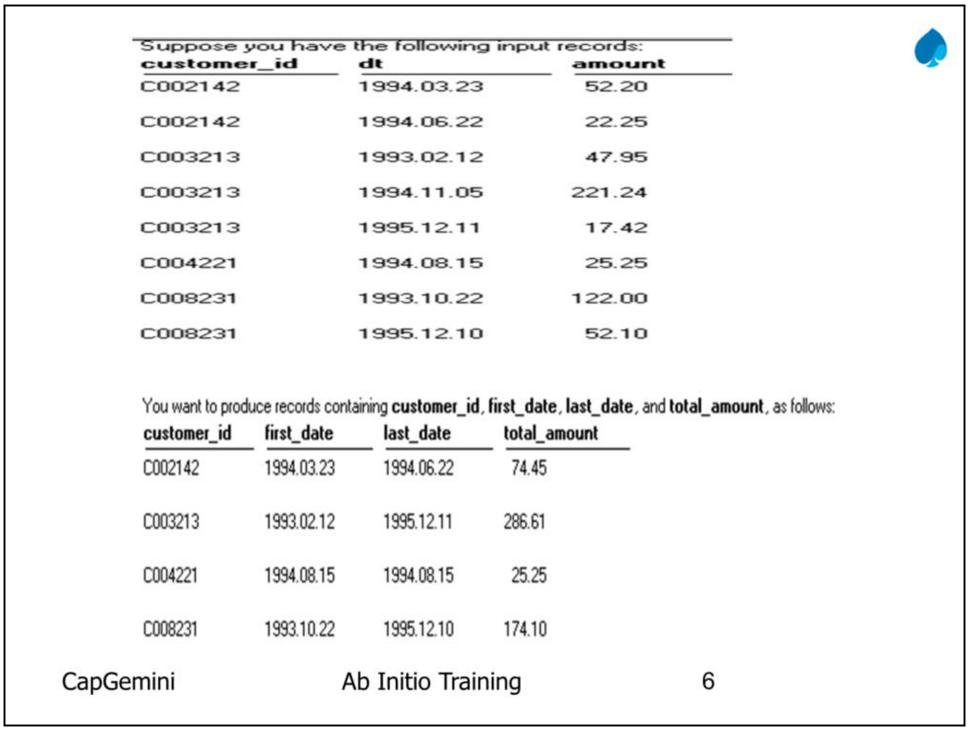


Alphabetical List of Rollup Parameters

- check-sort
- key
- key-method
- limit
- logging
- log_input
- log_intermediate
- log_output
- log_reject
- max-core
- ramp
- reject-threshold
- sorted-input
- transform

Graph Using a Rollup Component





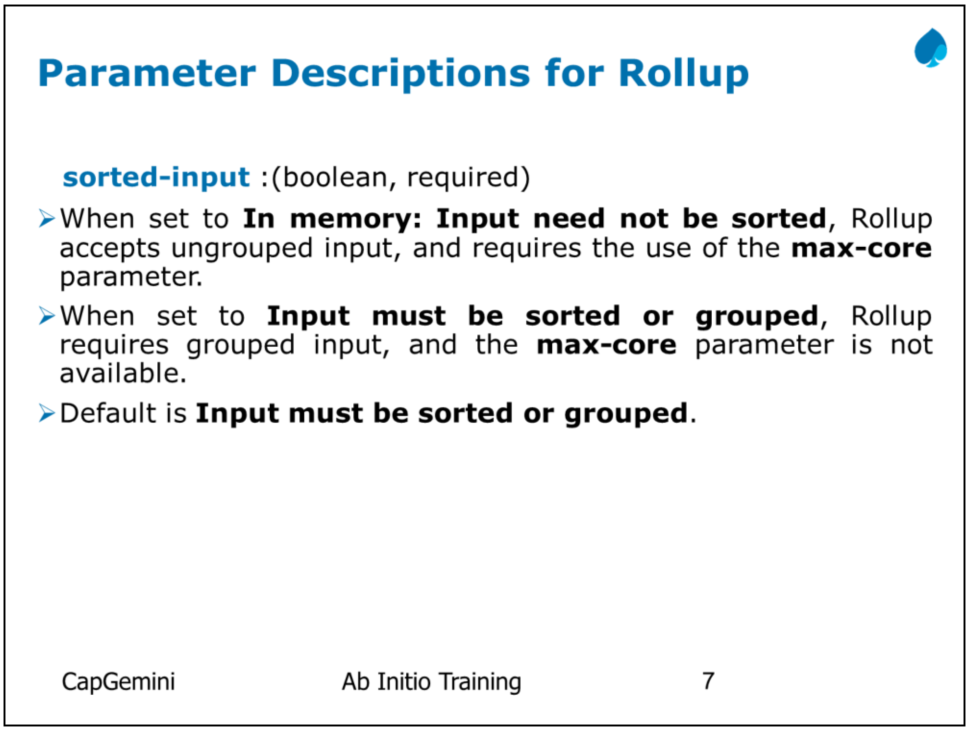
Suppose you have the following input records:

customer_id	dt	amount
C002142	1994.03.23	52.20
C002142	1994.06.22	22.25
C003213	1993.02.12	47.95
C003213	1994.11.05	221.24
C003213	1995.12.11	17.42
C004221	1994.08.15	25.25
C008231	1993.10.22	122.00
C008231	1995.12.10	52.10

You want to produce records containing **customer_id**, **first_date**, **last_date**, and **total_amount**, as follows:

customer_id	first_date	last_date	total_amount
C002142	1994.03.23	1994.06.22	74.45
C003213	1993.02.12	1995.12.11	286.61
C004221	1994.08.15	1994.08.15	25.25
C008231	1993.10.22	1995.12.10	174.10

CapGemini Ab Initio Training 6

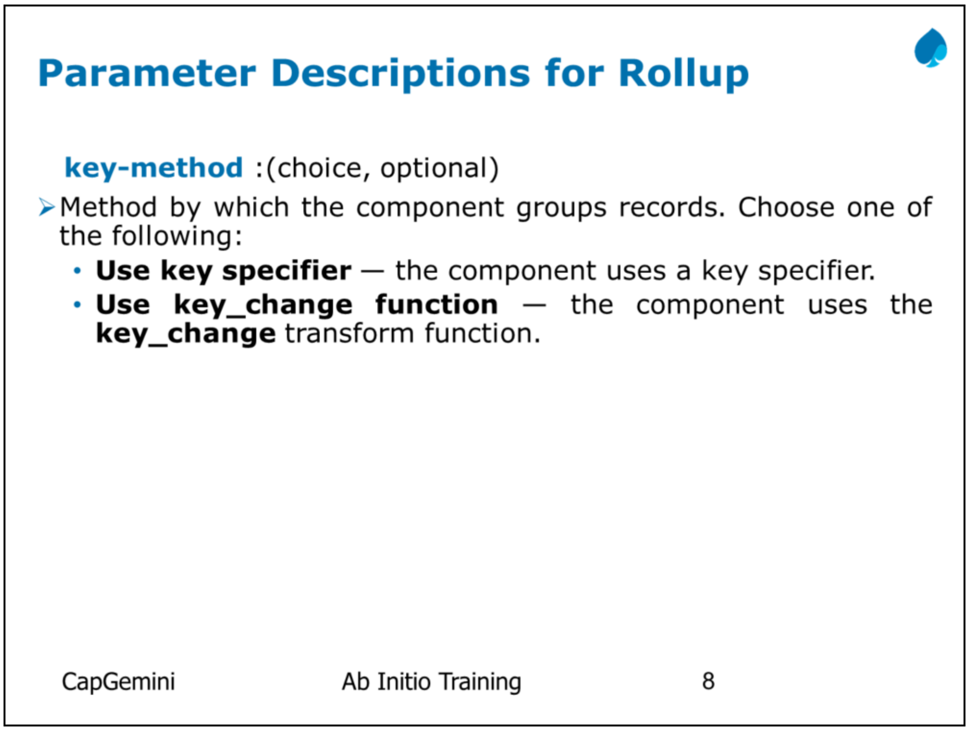


Parameter Descriptions for Rollup

sorted-input :(boolean, required)

- When set to **In memory: Input need not be sorted**, Rollup accepts ungrouped input, and requires the use of the **max-core** parameter.
- When set to **Input must be sorted or grouped**, Rollup requires grouped input, and the **max-core** parameter is not available.
- Default is **Input must be sorted or grouped**.

CapGemini Ab Initio Training 7



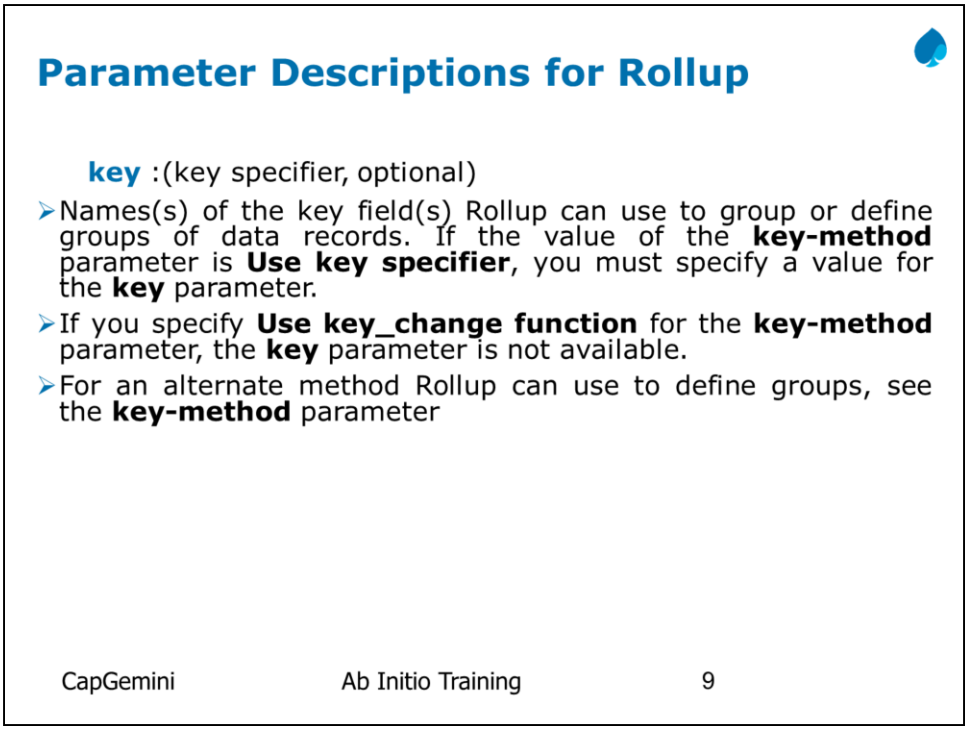
Parameter Descriptions for Rollup

key-method :(choice, optional)

➢ Method by which the component groups records. Choose one of the following:

- **Use key specifier** — the component uses a key specifier.
- **Use key_change function** — the component uses the **key_change** transform function.

CapGemini Ab Initio Training 8

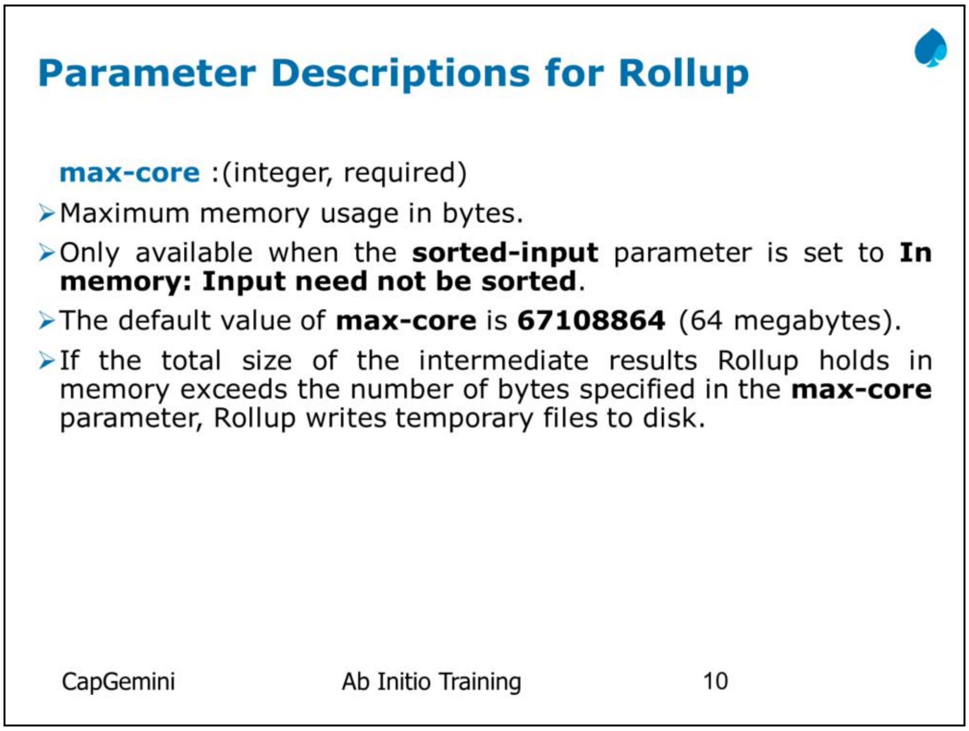


Parameter Descriptions for Rollup

key :(key specifier, optional)

- Names(s) of the key field(s) Rollup can use to group or define groups of data records. If the value of the **key-method** parameter is **Use key specifier**, you must specify a value for the **key** parameter.
- If you specify **Use key_change function** for the **key-method** parameter, the **key** parameter is not available.
- For an alternate method Rollup can use to define groups, see the **key-method** parameter

CapGemini Ab Initio Training 9

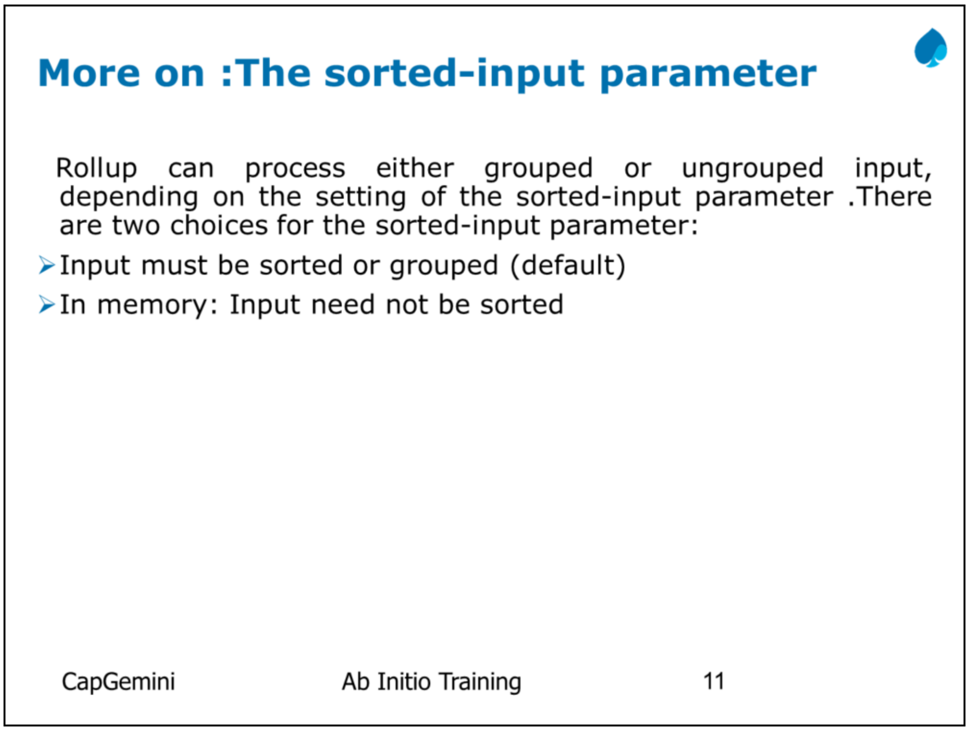


Parameter Descriptions for Rollup

max-core :(integer, required)

- Maximum memory usage in bytes.
- Only available when the **sorted-input** parameter is set to **In memory: Input need not be sorted**.
- The default value of **max-core** is **67108864** (64 megabytes).
- If the total size of the intermediate results Rollup holds in memory exceeds the number of bytes specified in the **max-core** parameter, Rollup writes temporary files to disk.

CapGemini Ab Initio Training 10

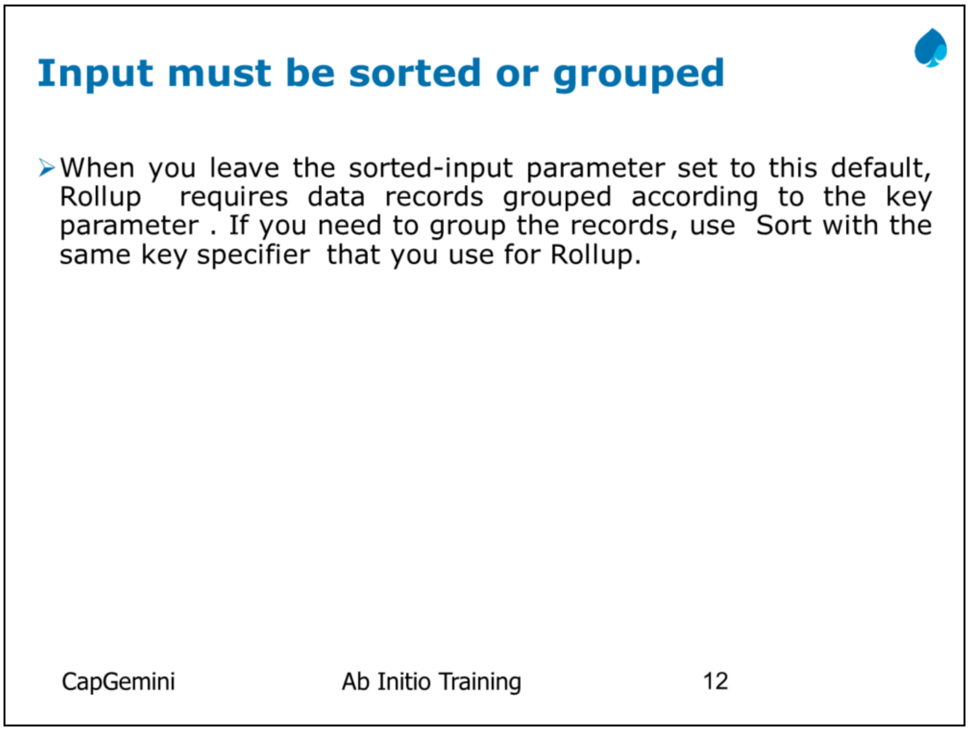


More on :The sorted-input parameter

Rollup can process either grouped or ungrouped input, depending on the setting of the sorted-input parameter .There are two choices for the sorted-input parameter:

- Input must be sorted or grouped (default)
- In memory: Input need not be sorted

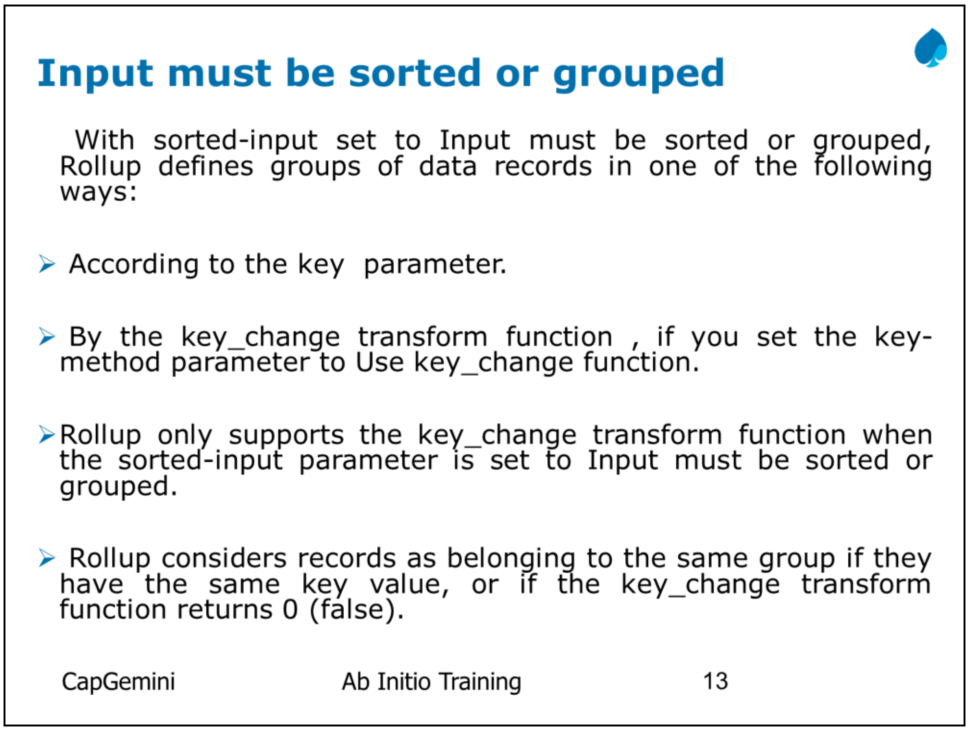
CapGemini Ab Initio Training 11



Input must be sorted or grouped

- When you leave the sorted-input parameter set to this default, Rollup requires data records grouped according to the key parameter . If you need to group the records, use Sort with the same key specifier that you use for Rollup.

CapGemini Ab Initio Training 12

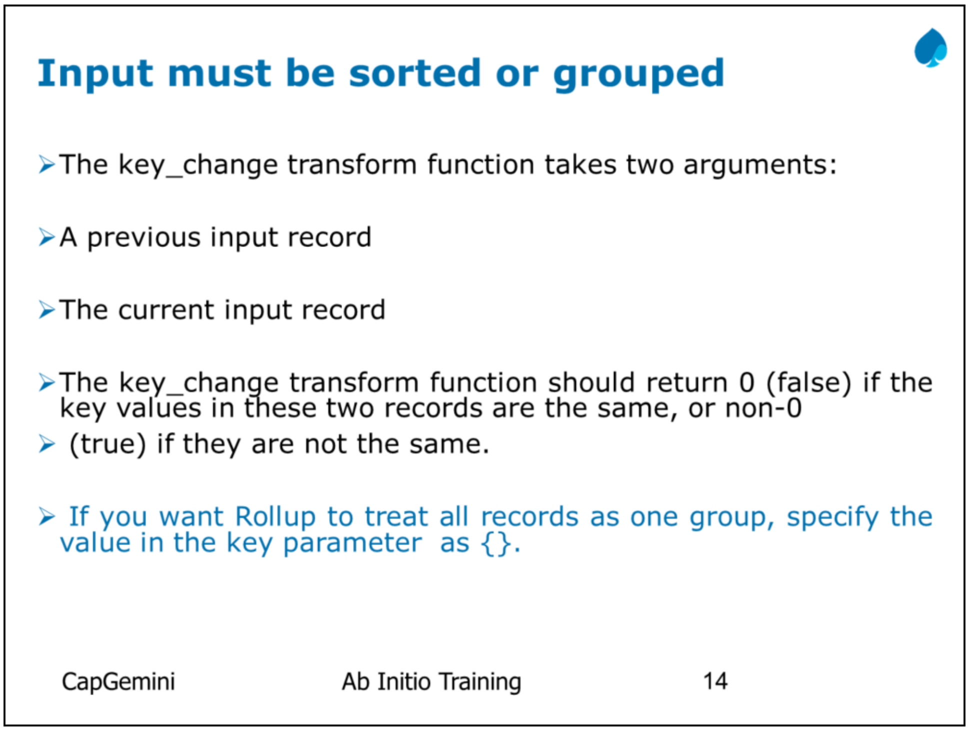


Input must be sorted or grouped

With sorted-input set to Input must be sorted or grouped, Rollup defines groups of data records in one of the following ways:

- According to the key parameter.
- By the key_change transform function , if you set the key-method parameter to Use key_change function.
- Rollup only supports the key_change transform function when the sorted-input parameter is set to Input must be sorted or grouped.
- Rollup considers records as belonging to the same group if they have the same key value, or if the key_change transform function returns 0 (false).

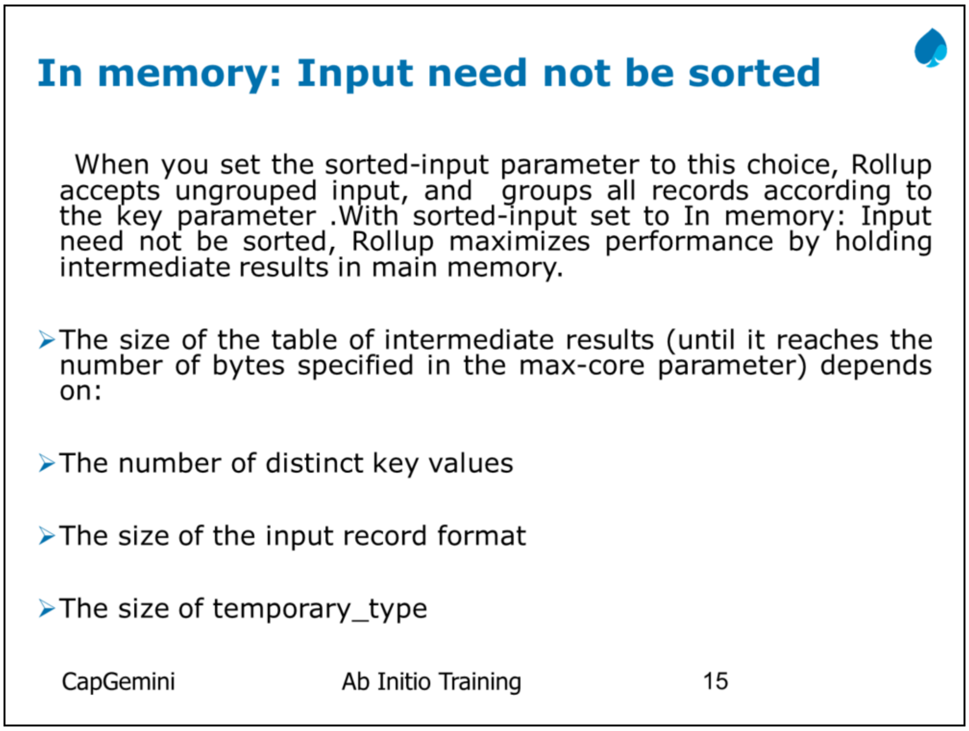
CapGemini Ab Initio Training 13



Input must be sorted or grouped

- The key_change transform function takes two arguments:
- A previous input record
- The current input record
- The key_change transform function should return 0 (false) if the key values in these two records are the same, or non-0
- (true) if they are not the same.
- If you want Rollup to treat all records as one group, specify the value in the key parameter as {}.

CapGemini Ab Initio Training 14

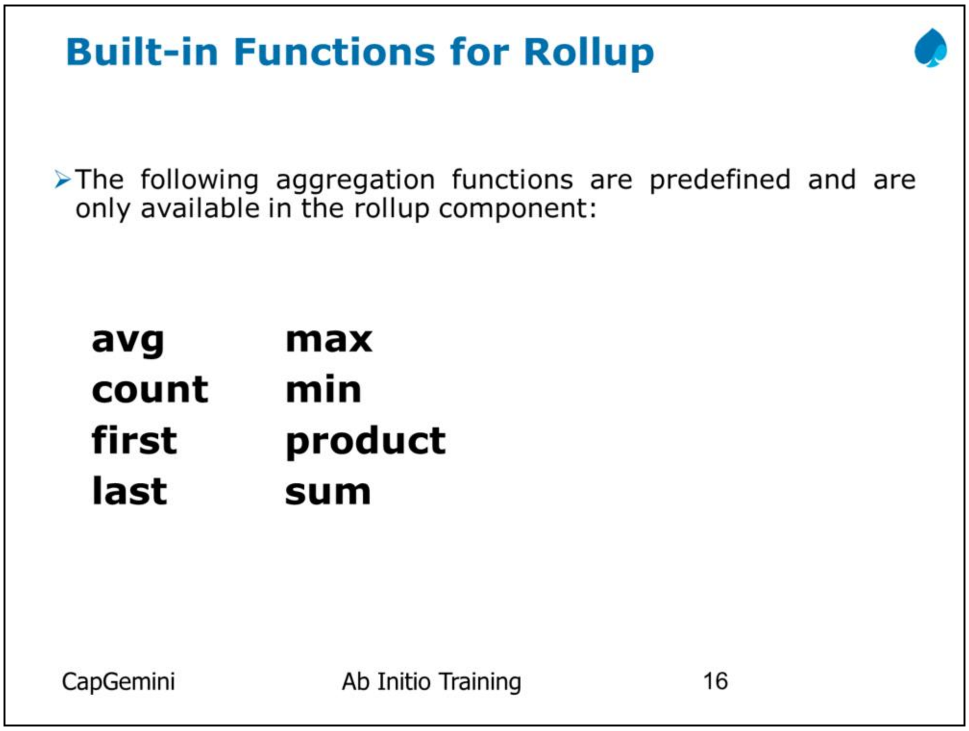


In memory: Input need not be sorted

When you set the sorted-input parameter to this choice, Rollup accepts ungrouped input, and groups all records according to the key parameter .With sorted-input set to In memory: Input need not be sorted, Rollup maximizes performance by holding intermediate results in main memory.

- The size of the table of intermediate results (until it reaches the number of bytes specified in the max-core parameter) depends on:
- The number of distinct key values
- The size of the input record format
- The size of temporary_type

CapGemini Ab Initio Training 15

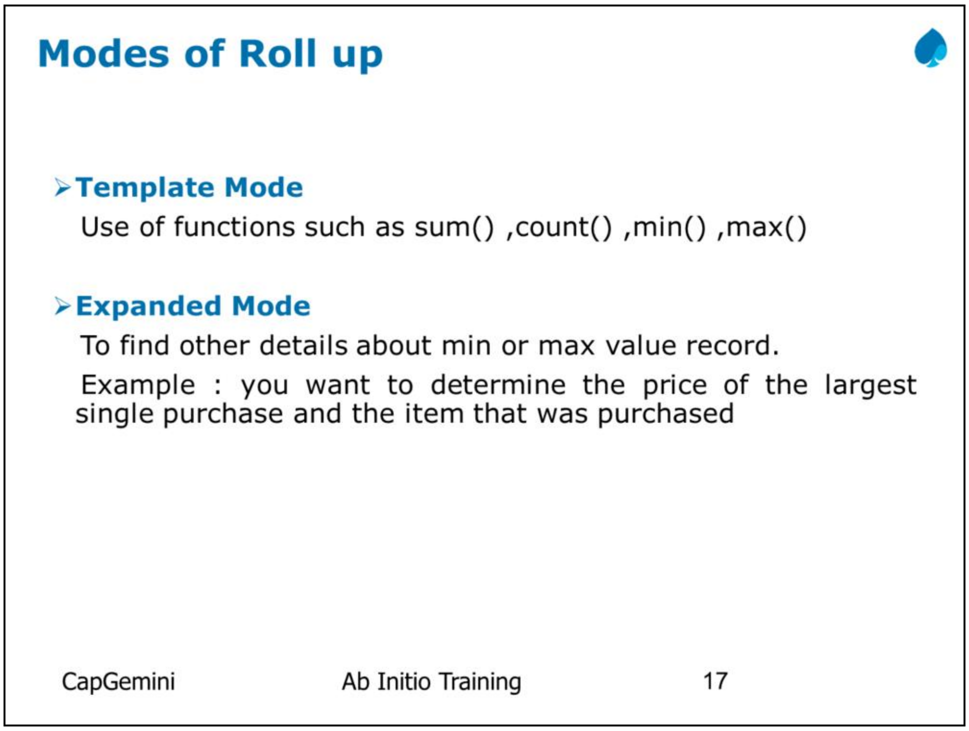


Built-in Functions for Rollup

- The following aggregation functions are predefined and are only available in the rollup component:

avg	**max**
count	**min**
first	**product**
last	**sum**

CapGemini Ab Initio Training 16



Modes of Roll up

- **Template Mode**

  Use of functions such as sum() ,count() ,min() ,max()

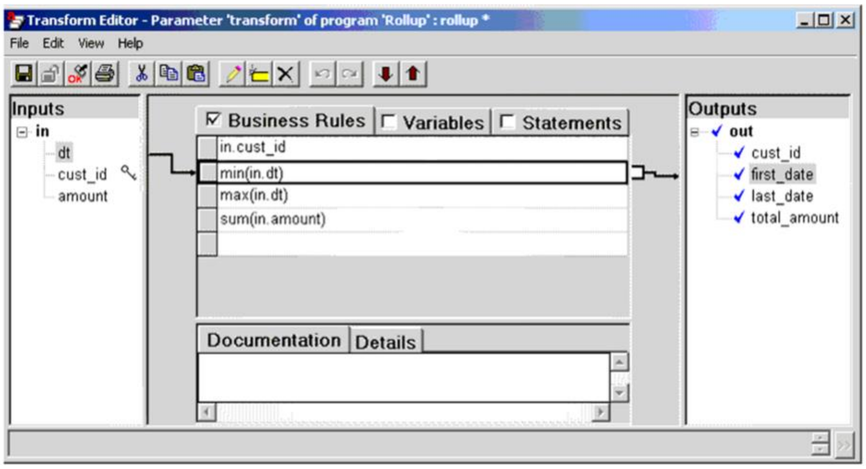
- **Expanded Mode**

  To find other details about min or max value record.

  Example : you want to determine the price of the largest single purchase and the item that was purchased

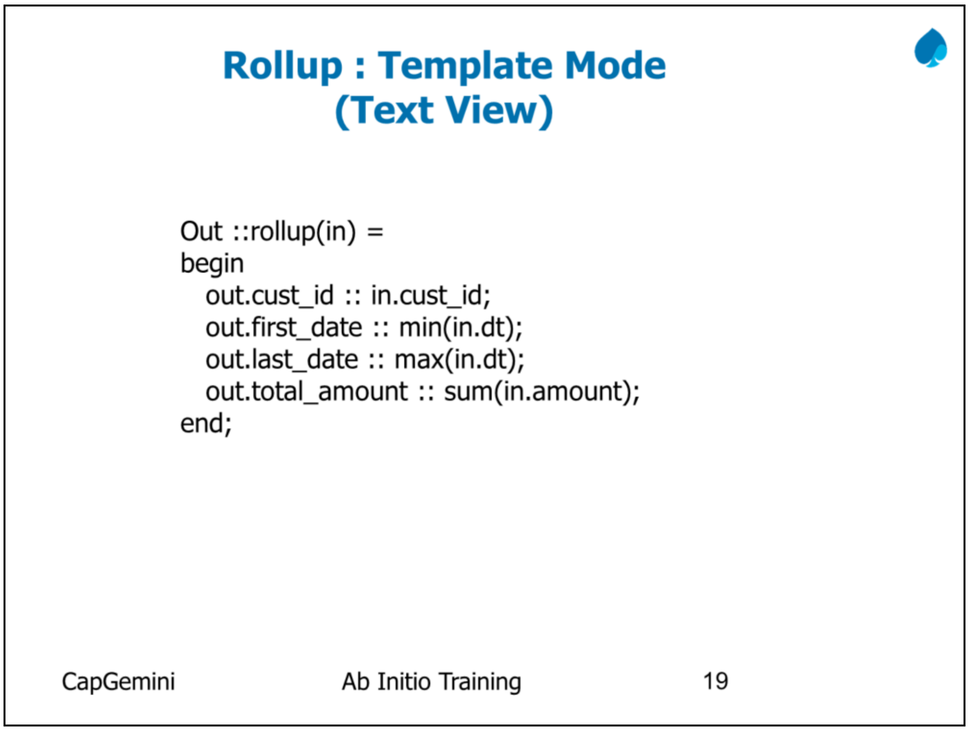
CapGemini Ab Initio Training 17

Rollup : Template Mode (Grid View)


Transform Editor - Parameter 'transform' of program 'Rollup' : rollup *

in.cust_id
min(in.dt)
max(in.dt)
sum(in.amount)

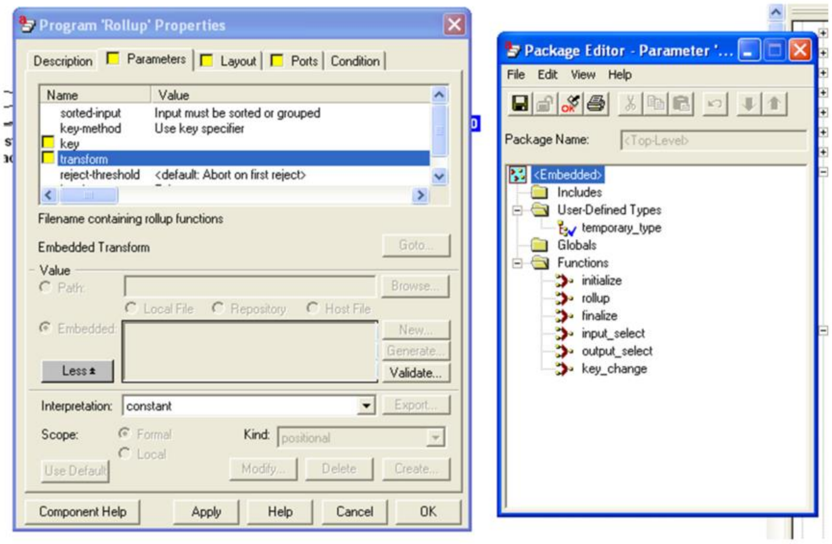
CapGemini Ab Initio Training 18



Rollup : Template Mode (Text View)

```
Out ::rollup(in) =
begin
   out.cust_id :: in.cust_id;
   out.first_date :: min(in.dt);
   out.last_date :: max(in.dt);
   out.total_amount :: sum(in.amount);
end;
```

Rollup Properties & Package Editor



Program 'Rollup' Properties

Description | Parameters | Layout | Ports | Condition

Name	Value
sorted-input	Input must be sorted or grouped
key-method	Use key specifier
key	
transform	
reject-threshold	<default: Abort on first reject>

Filename containing rollup functions

Embedded Transform

Value

Path:

Local File Repository Host File

Embedded:

Interpretation: constant

Scope: Formal Local

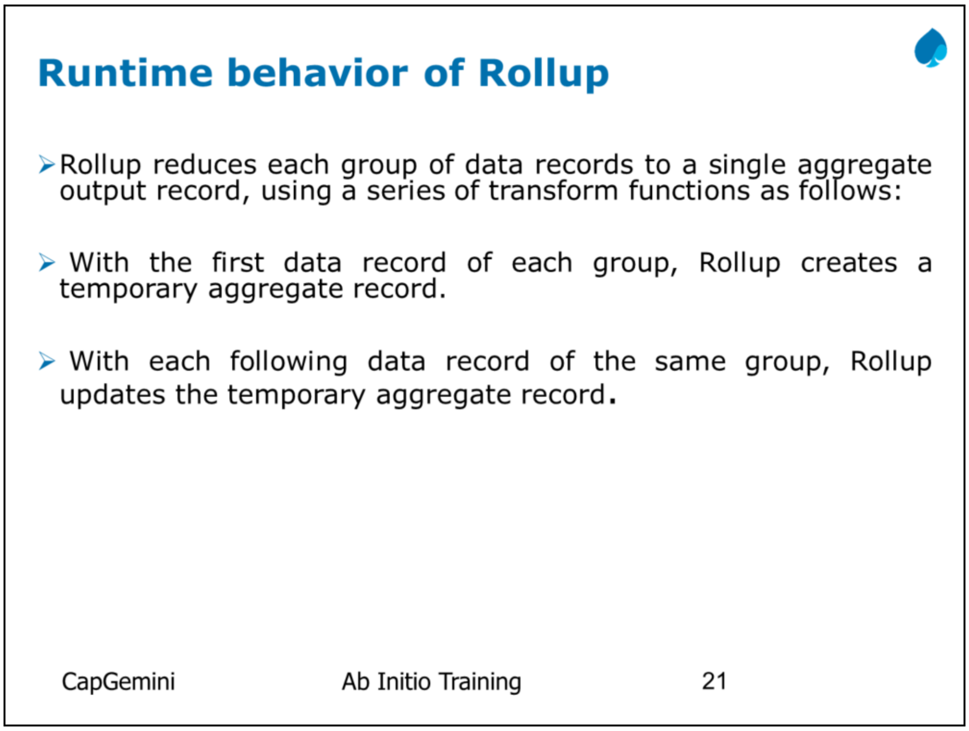
Kind: positional

Package Editor - Parameter '...

Package Name: <Top-Level>

```
<Embedded>
  Includes
  User-Defined Types
    temporary_type
  Globals
  Functions
    initialize
    rollup
    finalize
    input_select
    output_select
    key_change
```

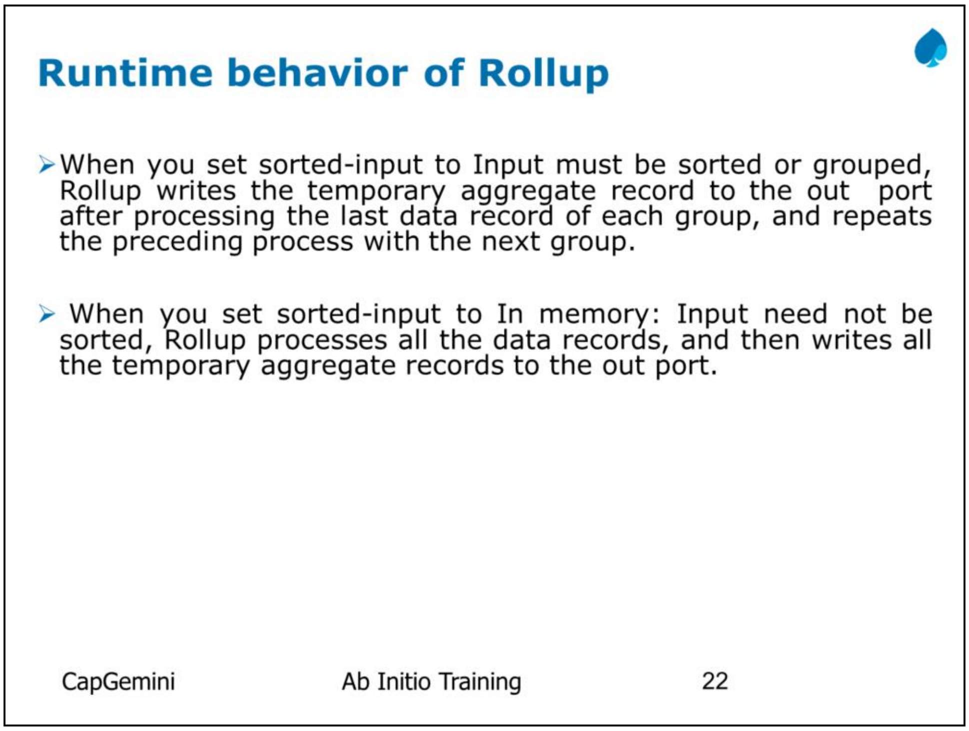
CapGemini Ab Initio Training 20



Runtime behavior of Rollup

- Rollup reduces each group of data records to a single aggregate output record, using a series of transform functions as follows:
- With the first data record of each group, Rollup creates a temporary aggregate record.
- With each following data record of the same group, Rollup updates the temporary aggregate record.

CapGemini Ab Initio Training 21

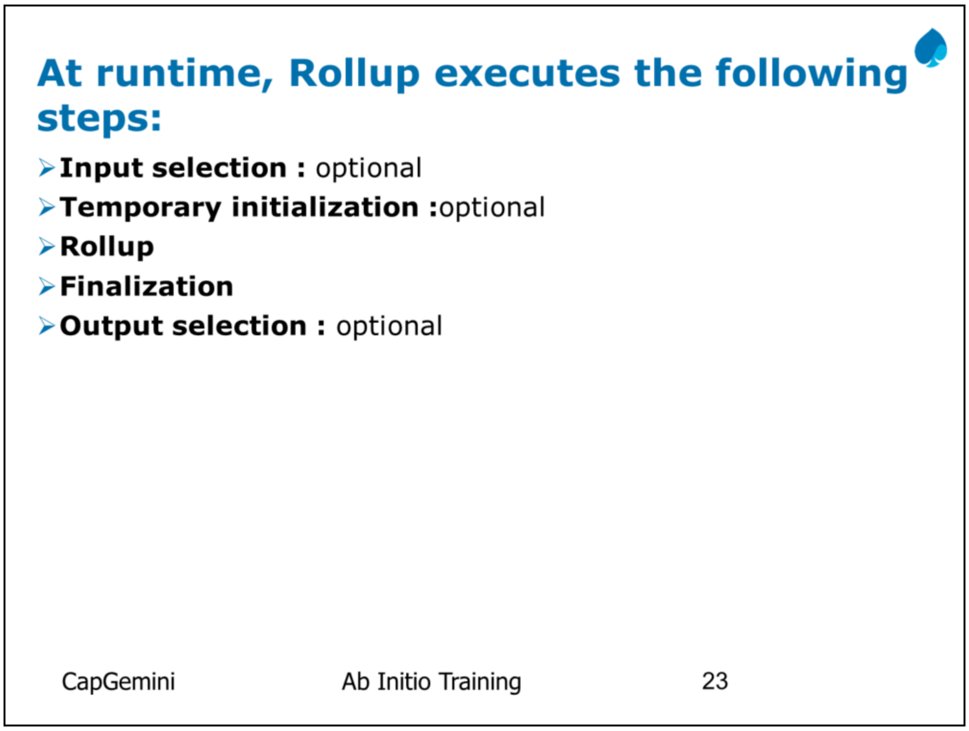


Runtime behavior of Rollup

- When you set sorted-input to Input must be sorted or grouped, Rollup writes the temporary aggregate record to the out port after processing the last data record of each group, and repeats the preceding process with the next group.

- When you set sorted-input to In memory: Input need not be sorted, Rollup processes all the data records, and then writes all the temporary aggregate records to the out port.

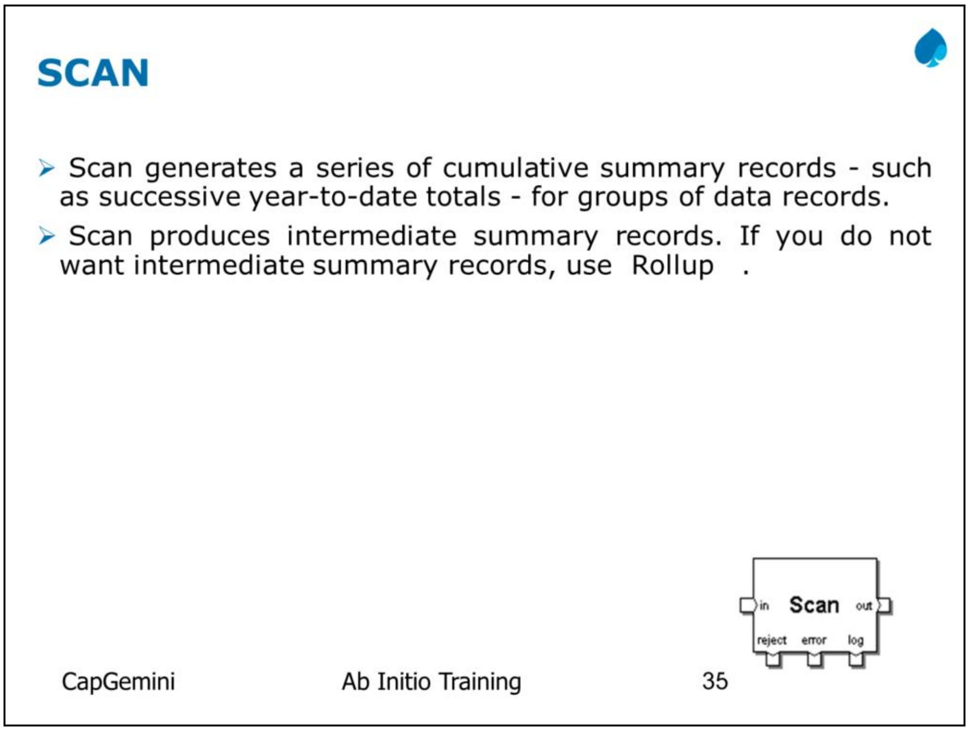
CapGemini Ab Initio Training 22



At runtime, Rollup executes the following steps:

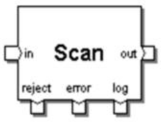
- **Input selection :** optional
- **Temporary initialization :**optional
- **Rollup**
- **Finalization**
- **Output selection :** optional

CapGemini Ab Initio Training 23

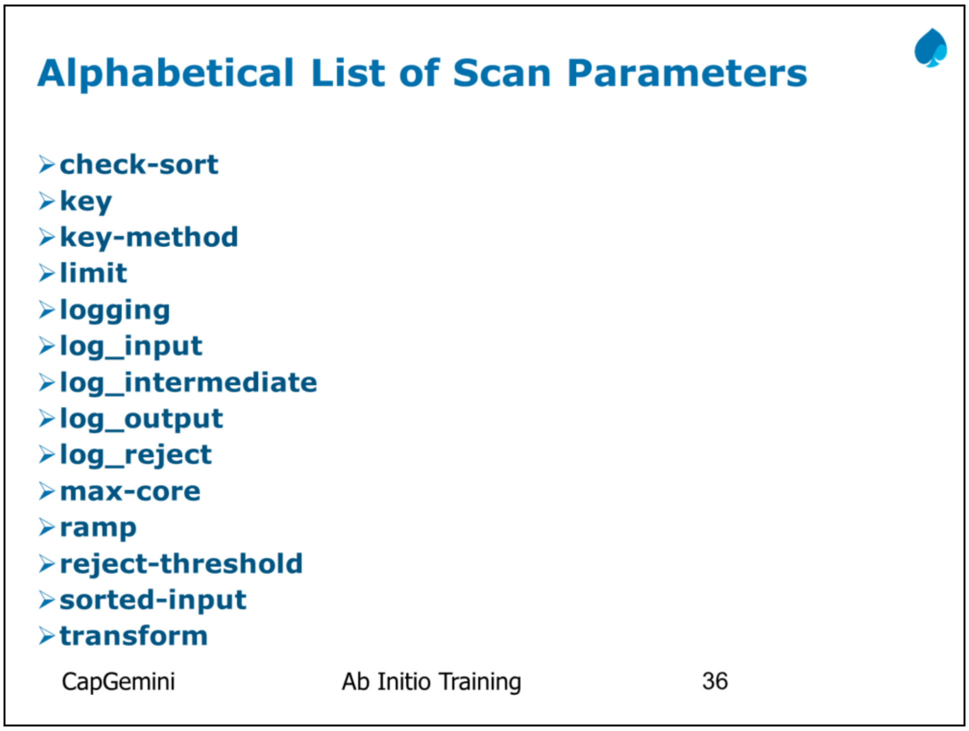


SCAN

- Scan generates a series of cumulative summary records - such as successive year-to-date totals - for groups of data records.
- Scan produces intermediate summary records. If you do not want intermediate summary records, use Rollup .



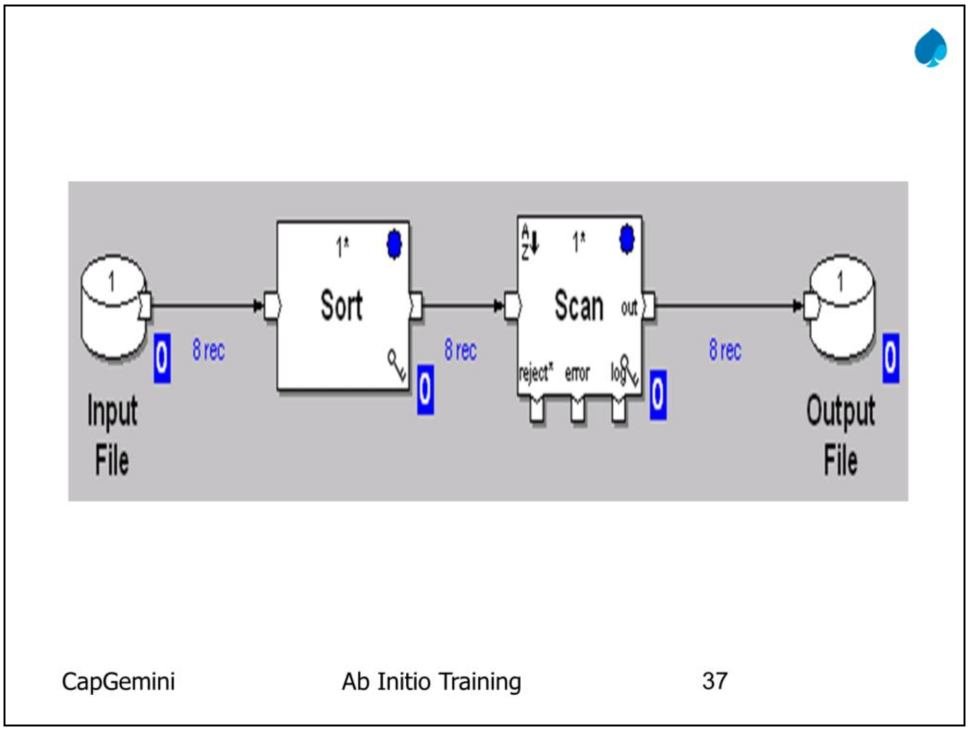
CapGemini Ab Initio Training 35



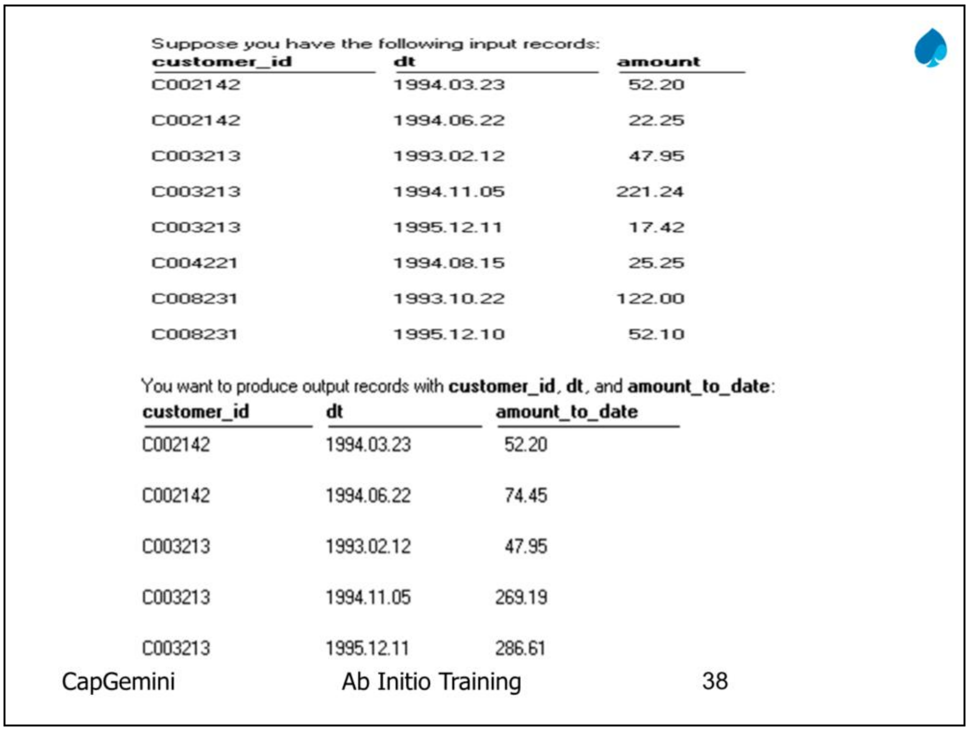
Alphabetical List of Scan Parameters

- check-sort
- key
- key-method
- limit
- logging
- log_input
- log_intermediate
- log_output
- log_reject
- max-core
- ramp
- reject-threshold
- sorted-input
- transform

CapGemini Ab Initio Training 36



CapGemini Ab Initio Training 37



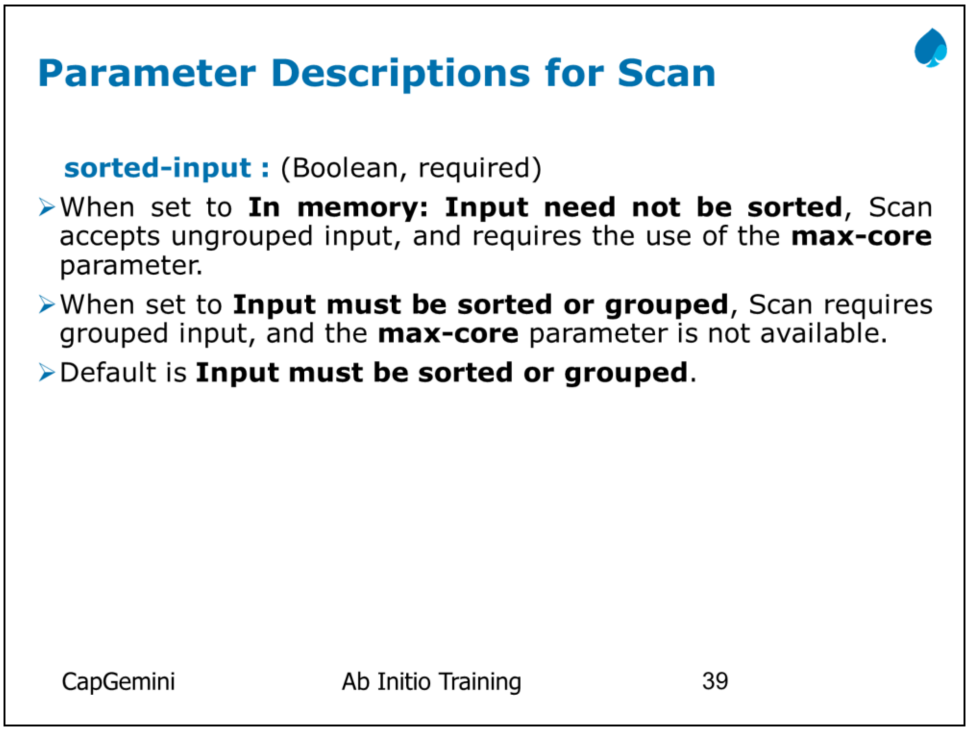
Suppose you have the following input records:

customer_id	dt	amount
C002142	1994.03.23	52.20
C002142	1994.06.22	22.25
C003213	1993.02.12	47.95
C003213	1994.11.05	221.24
C003213	1995.12.11	17.42
C004221	1994.08.15	25.25
C008231	1993.10.22	122.00
C008231	1995.12.10	52.10

You want to produce output records with **customer_id**, **dt**, and **amount_to_date**:

customer_id	dt	amount_to_date
C002142	1994.03.23	52.20
C002142	1994.06.22	74.45
C003213	1993.02.12	47.95
C003213	1994.11.05	269.19
C003213	1995.12.11	286.61

CapGemini Ab Initio Training 38

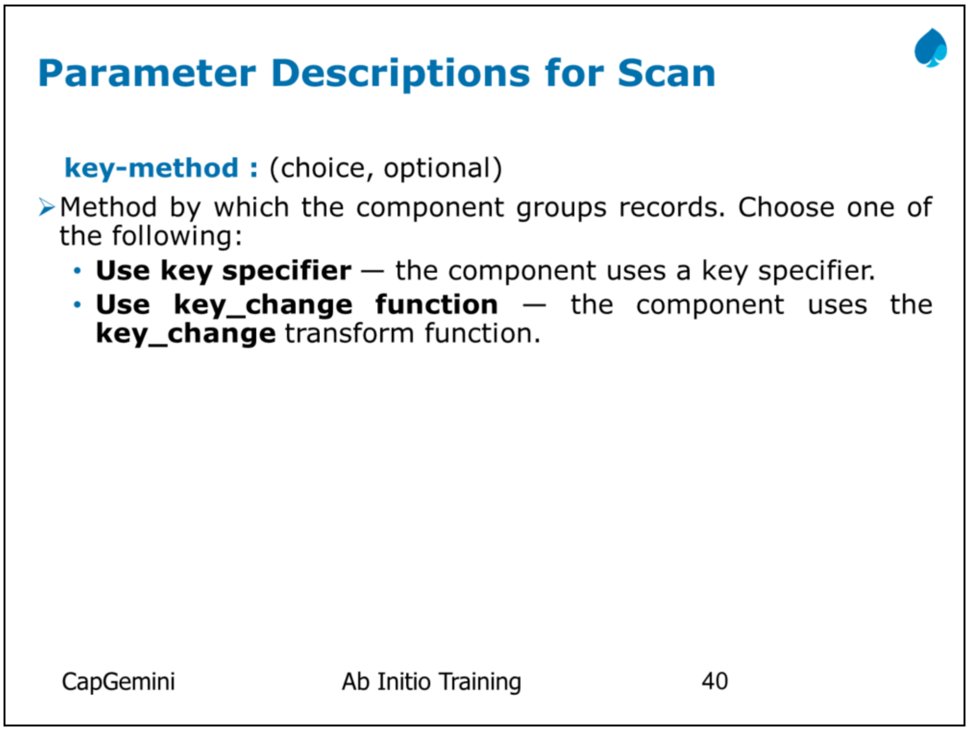


Parameter Descriptions for Scan

sorted-input : (Boolean, required)

- When set to **In memory: Input need not be sorted**, Scan accepts ungrouped input, and requires the use of the **max-core** parameter.
- When set to **Input must be sorted or grouped**, Scan requires grouped input, and the **max-core** parameter is not available.
- Default is **Input must be sorted or grouped**.

CapGemini Ab Initio Training 39

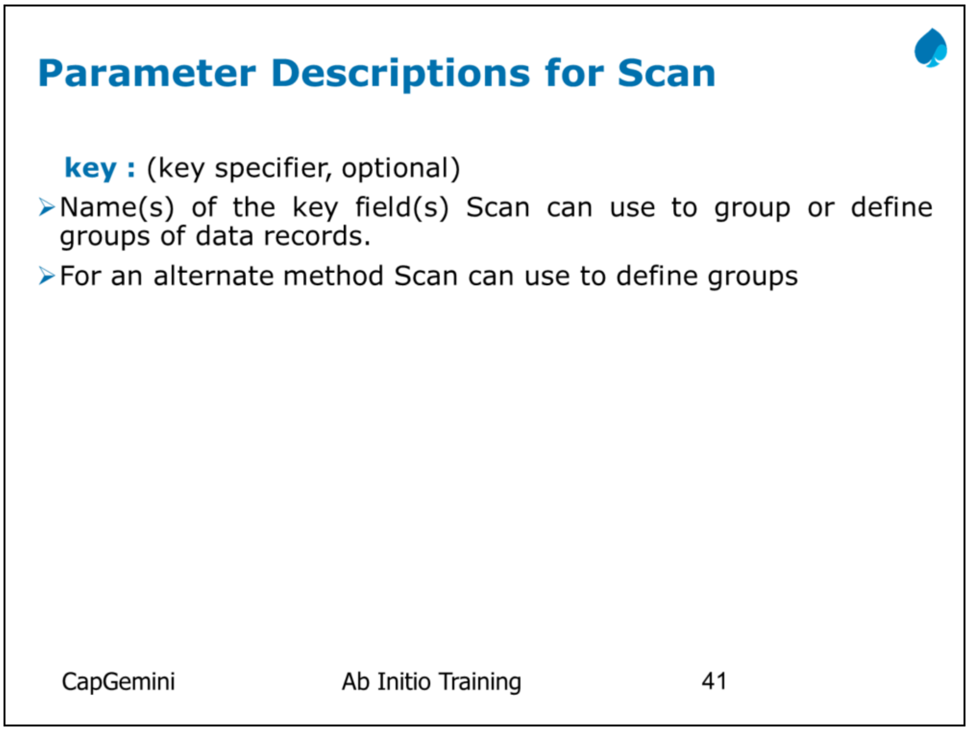


Parameter Descriptions for Scan

key-method : (choice, optional)

- Method by which the component groups records. Choose one of the following:
  - **Use key specifier** — the component uses a key specifier.
  - **Use key_change function** — the component uses the **key_change** transform function.

CapGemini Ab Initio Training 40

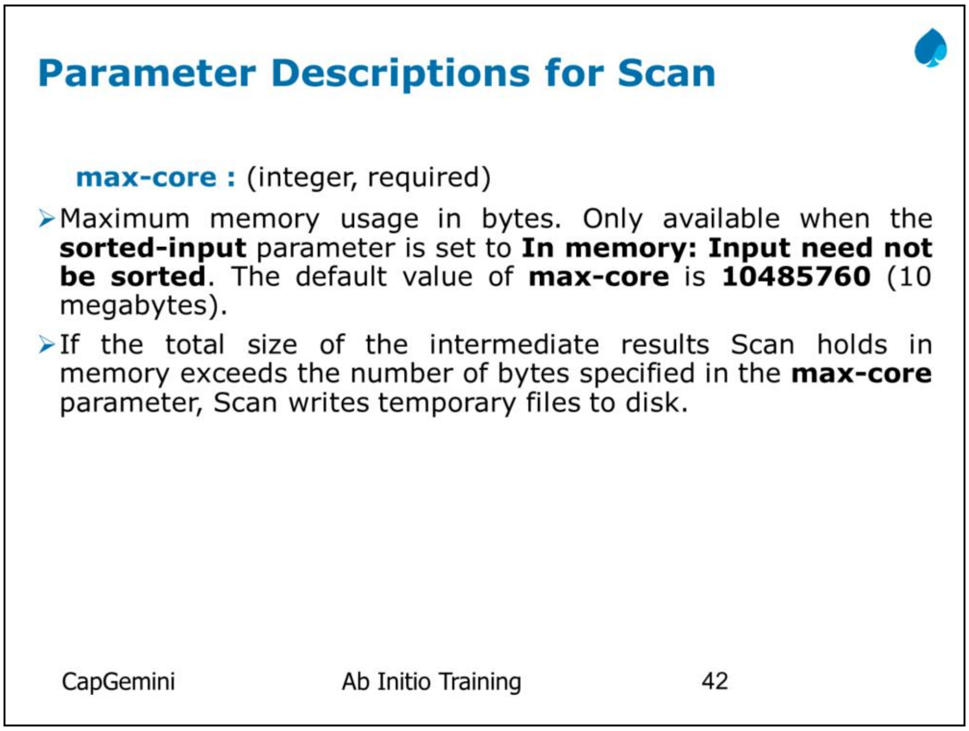


Parameter Descriptions for Scan

key : (key specifier, optional)

- Name(s) of the key field(s) Scan can use to group or define groups of data records.
- For an alternate method Scan can use to define groups

CapGemini Ab Initio Training 41

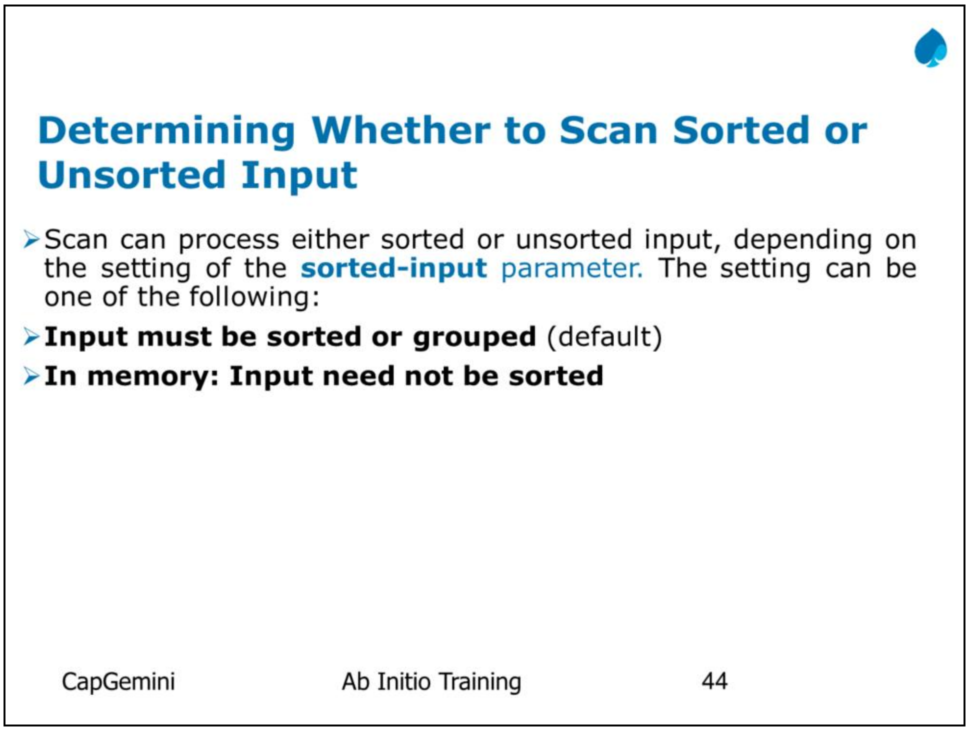


Parameter Descriptions for Scan

max-core : (integer, required)

- Maximum memory usage in bytes. Only available when the **sorted-input** parameter is set to **In memory: Input need not be sorted**. The default value of **max-core** is **10485760** (10 megabytes).
- If the total size of the intermediate results Scan holds in memory exceeds the number of bytes specified in the **max-core** parameter, Scan writes temporary files to disk.

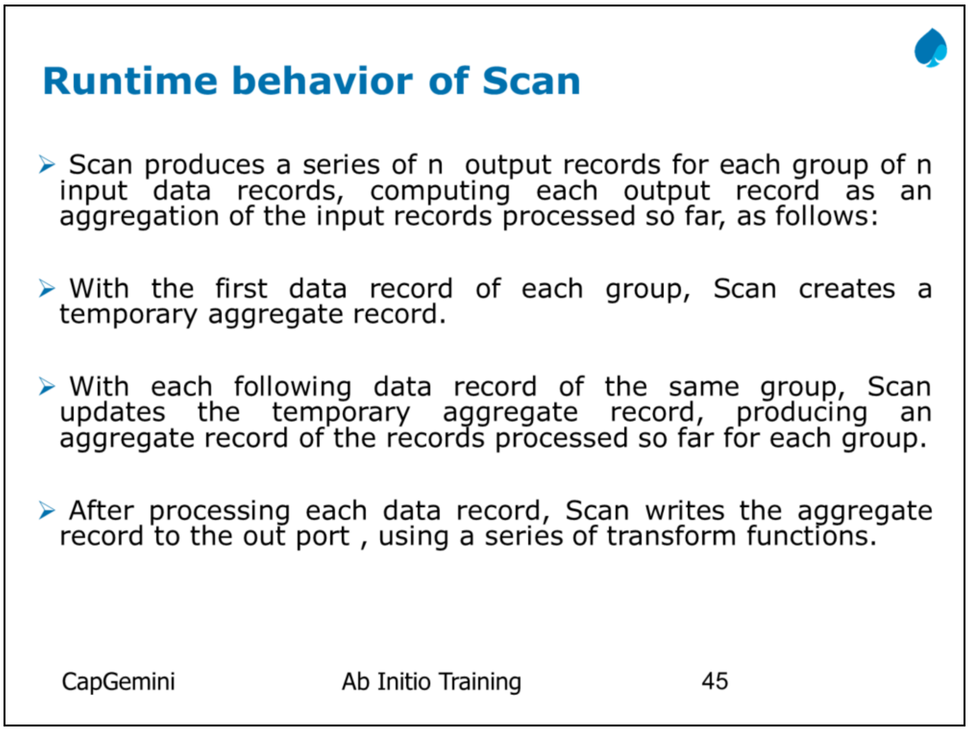
CapGemini Ab Initio Training 42



Determining Whether to Scan Sorted or Unsorted Input

- Scan can process either sorted or unsorted input, depending on the setting of the **sorted-input** parameter. The setting can be one of the following:
- **Input must be sorted or grouped** (default)
- **In memory: Input need not be sorted**

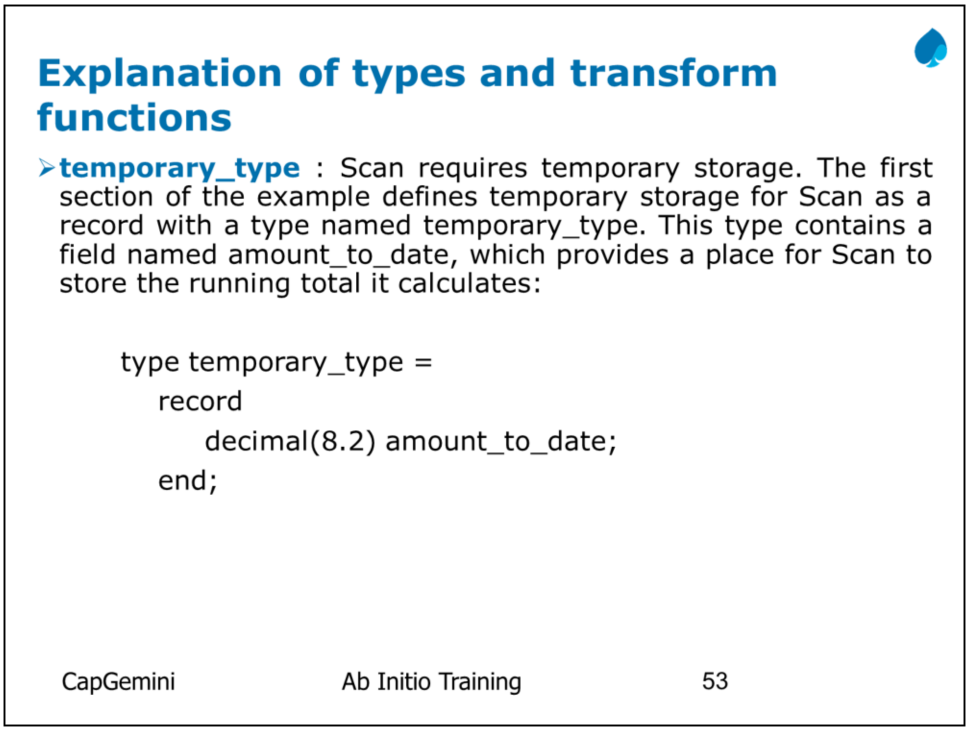
CapGemini Ab Initio Training 44



Runtime behavior of Scan

- Scan produces a series of n output records for each group of n input data records, computing each output record as an aggregation of the input records processed so far, as follows:
- With the first data record of each group, Scan creates a temporary aggregate record.
- With each following data record of the same group, Scan updates the temporary aggregate record, producing an aggregate record of the records processed so far for each group.
- After processing each data record, Scan writes the aggregate record to the out port , using a series of transform functions.

CapGemini Ab Initio Training 45

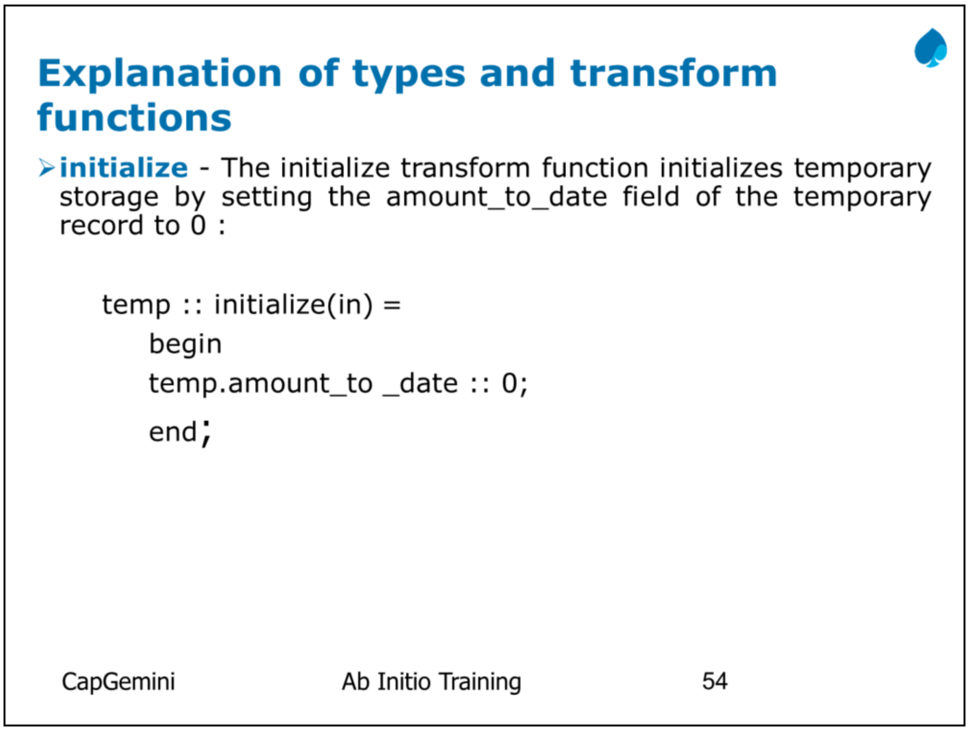


Explanation of types and transform functions

- **temporary_type** : Scan requires temporary storage. The first section of the example defines temporary storage for Scan as a record with a type named temporary_type. This type contains a field named amount_to_date, which provides a place for Scan to store the running total it calculates:

```
type temporary_type =
   record
       decimal(8.2) amount_to_date;
   end;
```

CapGemini Ab Initio Training 53

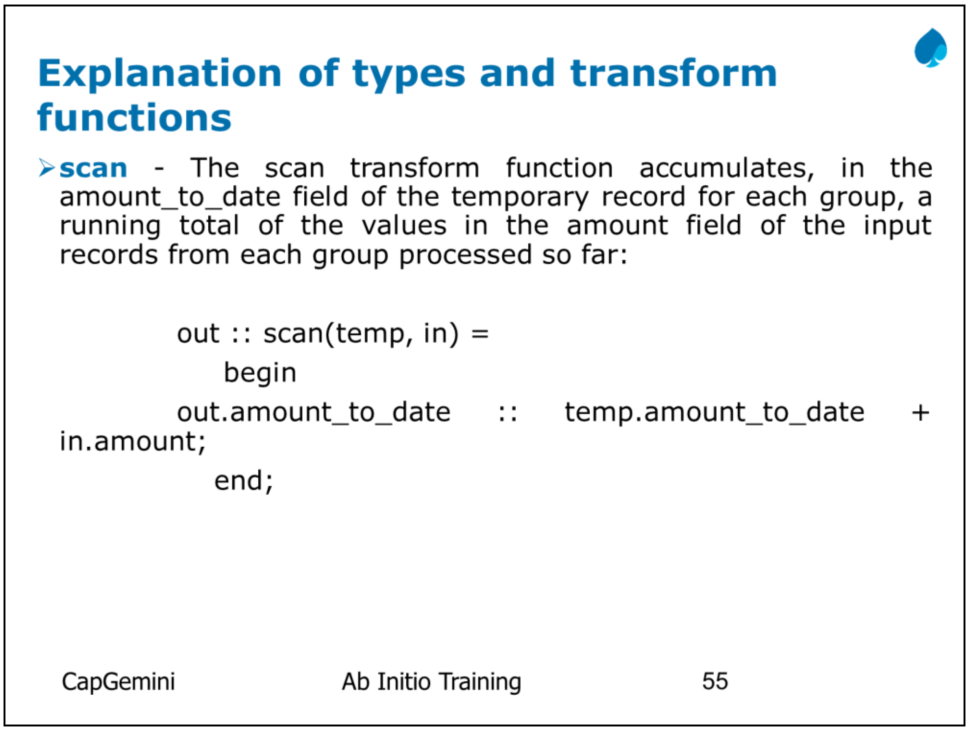


Explanation of types and transform functions

- **initialize** - The initialize transform function initializes temporary storage by setting the amount_to_date field of the temporary record to 0 :

```
temp :: initialize(in) =
    begin
    temp.amount_to _date :: 0;
    end;
```

CapGemini Ab Initio Training 54

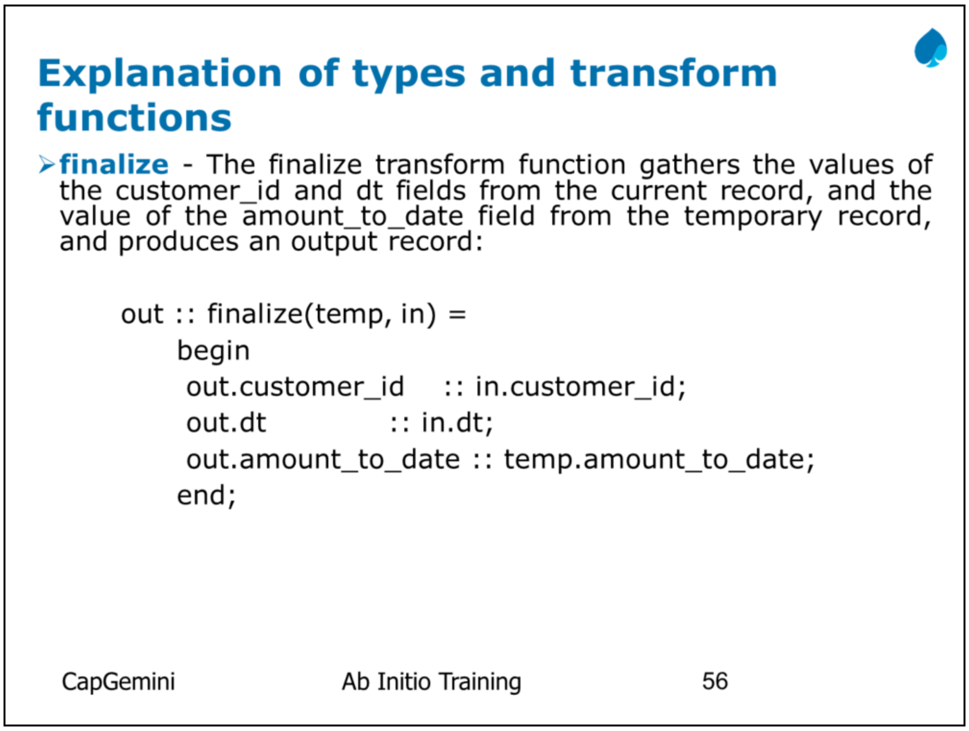


Explanation of types and transform functions

- **scan** - The scan transform function accumulates, in the amount_to_date field of the temporary record for each group, a running total of the values in the amount field of the input records from each group processed so far:

```
          out :: scan(temp, in) =
              begin
          out.amount_to_date    ::    temp.amount_to_date    +
in.amount;
             end;
```

CapGemini Ab Initio Training 55

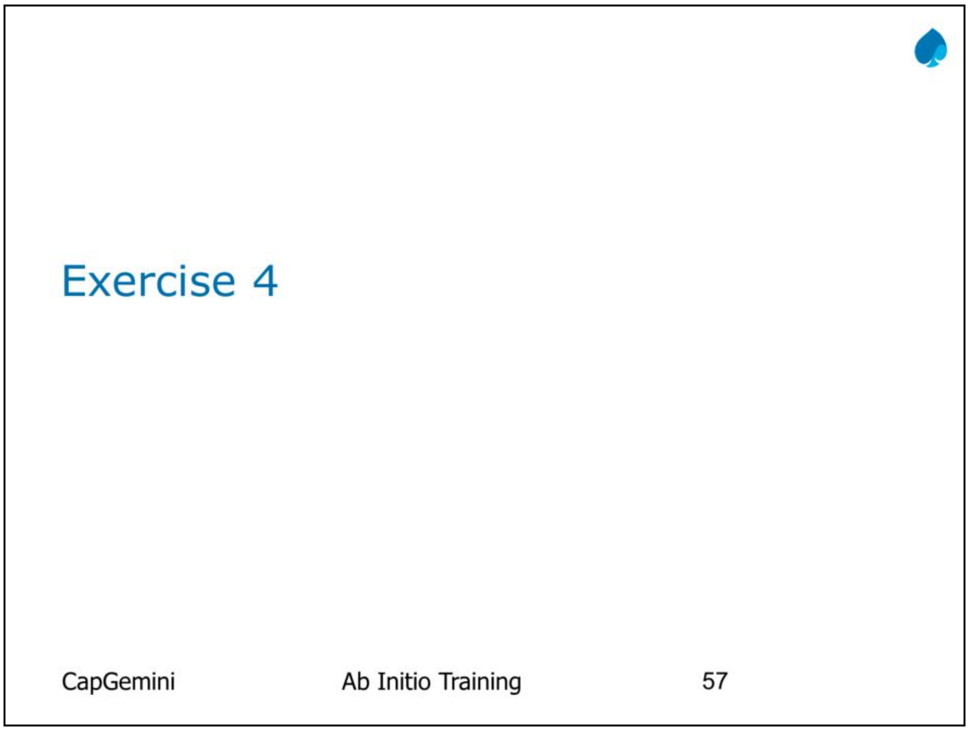


Explanation of types and transform functions

- **finalize** - The finalize transform function gathers the values of the customer_id and dt fields from the current record, and the value of the amount_to_date field from the temporary record, and produces an output record:

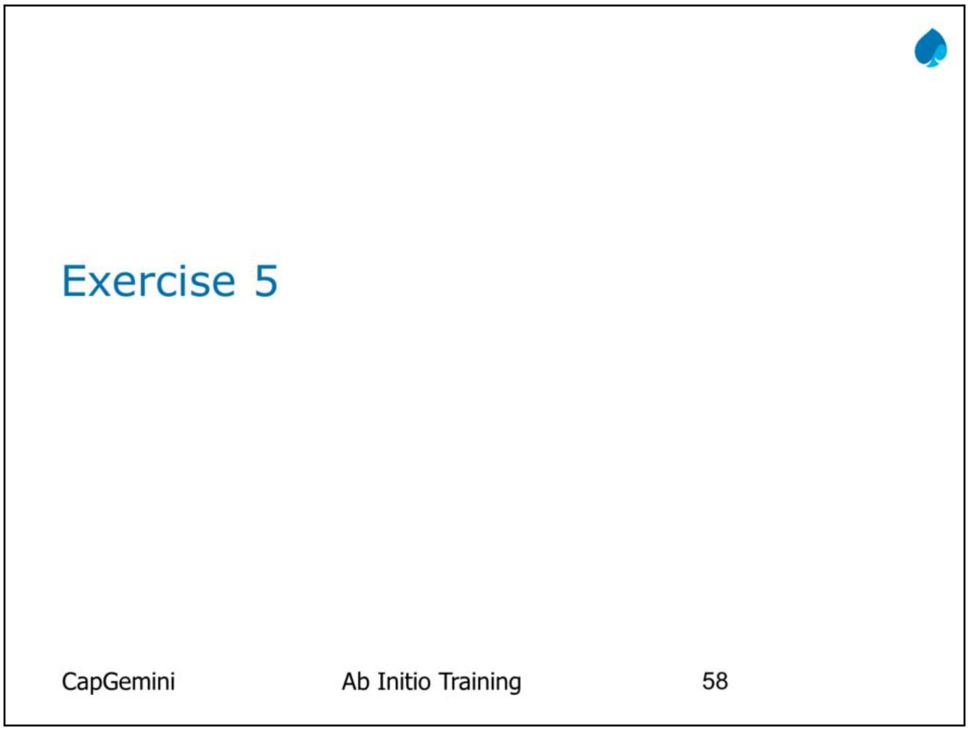
```
out :: finalize(temp, in) =
     begin
      out.customer_id     :: in.customer_id;
      out.dt              :: in.dt;
      out.amount_to_date :: temp.amount_to_date;
     end;
```

CapGemini Ab Initio Training 56



Exercise 4

CapGemini Ab Initio Training 57



Exercise 5

CapGemini Ab Initio Training 58

Thank You

End of Session 5

CapGemini Ab Initio Training 59