2.4 The New-Order Transaction

The New-Order business transaction consists of entering a complete order through a single database transaction. It represents a mid-weight, read-write transaction with a high frequency of execution and stringent response time requirements to satisfy on-line users. This transaction is the backbone of the workload. It is designed to place a variable load on the system to reflect on-line database activity as typically found in production environments.

2.4.1 Input Data Generation

2.4.1.1 For any given terminal, the home warehouse number (W_ID) is constant over the whole measurement interval (see Clause 5.5).

2.4.1.2 The district number (D_ID) is randomly selected within [1 .. 10] from the home warehouse (D_W_ID = W_ID). The non-uniform random customer number (C_ID) is selected using the NURand(1023,1,3000) function from the selected district number (C_D_ID = D_ID) and the home warehouse number (C_W_ID = W_ID).

2.4.1.3 The number of items in the order (*ol_cnt*) is randomly selected within [5 .. 15] (an average of 10). This field is not entered. It is generated by the terminal emulator to determine the size of the order. O_OL_CNT is later displayed after being computed by the SUT.

2.4.1.4 A fixed 1% of the New-Order transactions are chosen at random to simulate user data entry errors and exercise the performance of rolling back update transactions. This must be implemented by generating a random number *rbk* within [1 .. 100].

Comment: All New-Order transactions must have independently generated input data. The input data from a rolled back transaction cannot be used for a subsequent transaction.

2.4.1.5 For each of the *ol_cnt* items on the order:

1. A non-uniform random item number (OL_I_ID) is selected using the NURand(8191,1,100000) function. If this is the last item on the order and *rbk* = 1 (see Clause 2.4.1.4), then the item number is set to an unused value.

   Comment: An **unused** value for an item number is a value not found in the database such that its use will produce a "not-found" condition within the application program. This condition should result in rolling back the current database transaction.

2. A supplying warehouse number (OL_SUPPLY_W_ID) is selected as the home warehouse 99% of the time and as a remote warehouse 1% of the time. This can be implemented by generating a random number *x* within [1 .. 100];

   - If *x* > 1, the item is supplied from the home warehouse (OL_SUPPLY_W_ID = W_ID).

   - If *x* = 1, the item is supplied from a remote warehouse (OL_SUPPLY_W_ID is randomly selected within the range of active warehouses (see Clause 4.2.2) other than W_ID).

   Comment 1: With an average of 10 items per order, approximately 90% of all orders can be supplied in full by stocks from the home warehouse.

   Comment 2: If the system is configured for a single warehouse, then all items are supplied from that single home warehouse.

3. A quantity (OL_QUANTITY) is randomly selected within [1 .. 10].

2.4.1.6 The order entry date (O_ENTRY_D) is generated within the SUT by using the current system date and time.

2.4.1.7 An order-line is said to be **home** if it is supplied by the home warehouse (i.e., when OL_SUPPLY_W_ID equals O_W_ID).

2.4.1.8 An order-line is said to be **remote** when it is supplied by a remote warehouse (i.e., when OL_SUPPLY_W_ID does not equal O_W_ID).

2.4.2 Transaction Profile

2.4.2.1 Entering a new order is done in a single database transaction with the following steps:

1. Create an order header, comprised of:

   2 row selections with data retrieval,

   1 row selection with data retrieval and update,

   2 row insertions.

2. Order a variable number of items (average *ol_cnt* = 10), comprised of:

   (1 * *ol_cnt*) row selections with data retrieval,

   (1 * *ol_cnt*) row selections with data retrieval and update,

   (1 * *ol_cnt*) row insertions.

Note: The above summary is provided for information only. The actual requirement is defined by the detailed transaction profile below.

2.4.2.2 For a given warehouse number (W_ID), district number (D_W_ID , D_ID), customer number (C_W_ID , C_D_ID , C_ ID), count of items (*ol_cnt*, not communicated to the SUT), and for a given set of items (OL_I_ID), supplying warehouses (OL_SUPPLY_W_ID), and quantities (OL_QUANTITY):

- The input data (see Clause 2.4.3.2) are communicated to the SUT.
- A database transaction is started.
- The row in the WAREHOUSE table with matching W_ID is selected and W_TAX, the warehouse tax rate, is retrieved.
- The row in the DISTRICT table with matching D_W_ID and D_ ID is selected, D_TAX, the district tax rate, is retrieved, and D_NEXT_O_ID, the next available order number for the district, is retrieved and incremented by one.
- The row in the CUSTOMER table with matching C_W_ID, C_D_ID, and C_ID is selected and C_DISCOUNT, the customer's discount rate, C_LAST, the customer's last name, and C_CREDIT, the customer's credit status, are retrieved.
- A new row is inserted into both the NEW-ORDER table and the ORDER table to reflect the creation of the new order. O_CARRIER_ID is set to a null value. If the order includes only home order-lines, then O_ALL_LOCAL is set to 1, otherwise O_ALL_LOCAL is set to 0.
- The number of items, O_OL_CNT, is computed to match *ol_cnt*.

- For each O_OL_CNT item on the order:
  - The row in the ITEM table with matching I_ID (equals OL_I_ID) is selected and I_PRICE, the price of the item, I_NAME, the name of the item, and I_DATA are retrieved. If I_ID has an unused value (see Clause 2.4.1.5), a "not-found" condition is signaled, resulting in a rollback of the database transaction (see Clause 2.4.2.3).
  - The row in the STOCK table with matching S_I_ID (equals OL_I_ID) and S_W_ID (equals OL_SUPPLY_W_ID) is selected. S_QUANTITY, the quantity in stock, S_DIST_xx, where xx represents the district number, and S_DATA are_retrieved. If the retrieved value for S_QUANTITY exceeds OL_QUANTITY by 10 or more, then S_QUANTITY is decreased by OL_QUANTITY; otherwise S_QUANTITY is updated to (S_QUANTITY - OL_QUANTITY)+91. S_YTD is increased by OL_QUANTITY and S_ORDER_CNT is incremented by 1. If the order-line is remote, then S_REMOTE_CNT is incremented by 1.
  - The amount for the item in the order (OL_AMOUNT) is computed as:

    OL_QUANTITY * I_PRICE
  - The strings in I_DATA and S_DATA are examined. If they both include the string "ORIGINAL", the *brand-generic* field for that item is set to "B", otherwise, the *brand-generic* field is set to "G".
  - A new row is inserted into the ORDER-LINE table to reflect the item on the order. OL_DELIVERY_D is set to a null value, OL_NUMBER is set to a unique value within all the ORDER-LINE rows that have the same OL_O_ID value, and OL_DIST_INFO is set to the content of S_DIST_xx, where xx represents the district number (OL_D_ID)
- The *total-amount* for the complete order is computed as:

  sum(OL_AMOUNT) * (1 - C_DISCOUNT) * (1 + W_TAX + D_TAX)
- The database transaction is committed, unless it has been rolled back as a result of an *unused* value for the last item number (see Clause 2.4.1.5).
- The output data (see Clause 2.4.3.3) are communicated to the terminal.

2.4.2.3 For transactions that rollback as a result of an unused item number, the complete transaction profile must be executed with the exception that the following steps need not be done:

- Selecting and retrieving the row in the STOCK table with S_I_ID matching the unused item number.
- Examining the strings I_DATA and S_DATA for the unused item.
- Inserting a new row into the ORDER-LINE table for the unused item.
- Adding the amount for the unused item to the sum of all OL_AMOUNT.

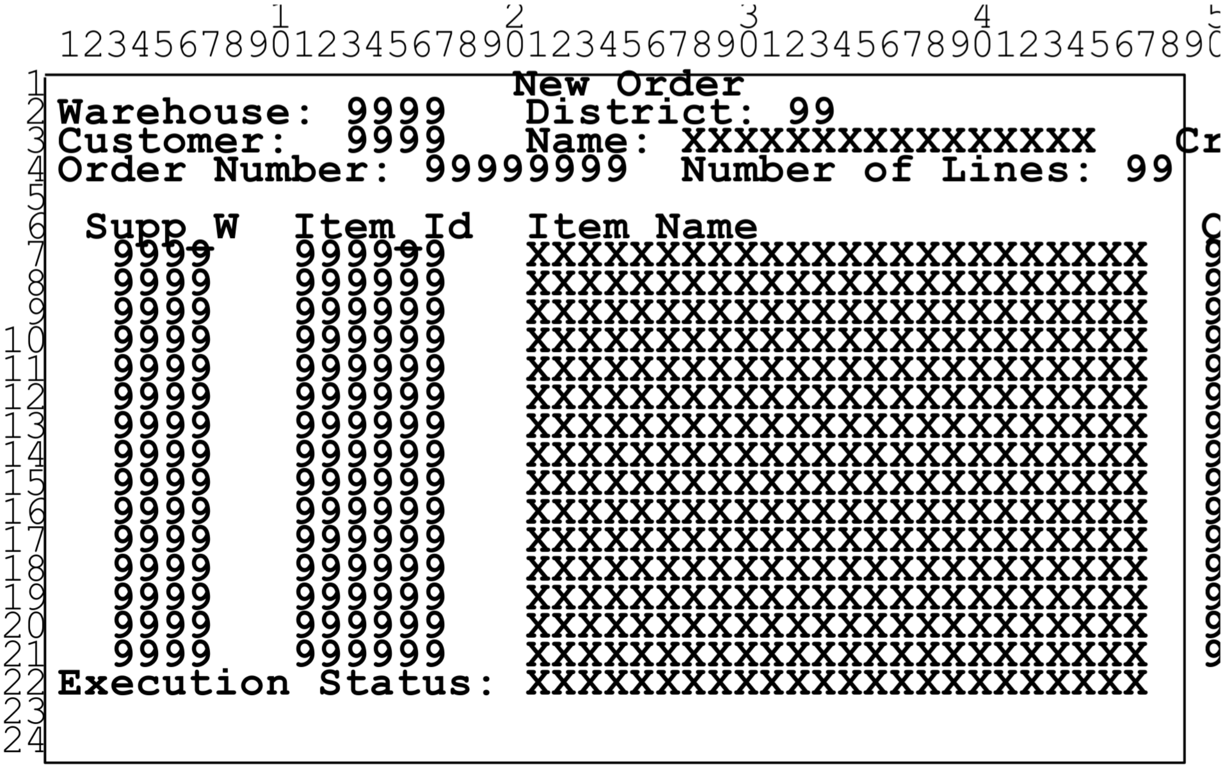
The transaction is not committed. Instead, the transaction is rolled back.

Comment 1: The intent of this clause is to ensure that within the New-Order transaction all valid items are processed prior to processing the unused item. Knowledge that an item is unused, resulting in rolling back the transaction, can only be used to skip execution of the above steps. No other optimization can result from this knowledge (e.g., skipping other steps, changing the execution of other steps, using a different type of transaction, etc.).

Comment 2: This clause is an exception to Clause 2.3.1. The order of data manipulations prior to signaling a "not found" condition is immaterial.

2.4.3 Terminal I/O

2.4.3.1 For each transaction the originating terminal must display the following input/ output screen with all input and output fields cleared (with either spaces or zeros) except for the Warehouse field which has not changed and must display the fixed W_ID value associated with that terminal.



```
            1         2         3         4         5
   12345678901234567890123456789012345678901234567890
 1                      New Order
 2 Warehouse: 9999   District: 99
 3 Customer:  9999   Name: XXXXXXXXXXXXXXXX   Cr
 4 Order Number: 99999999  Number of Lines: 99
 5
 6  Supp_W  Item_Id  Item Name                  C
 7    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
 8    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
 9    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
10    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
11    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
12    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
13    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
14    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
15    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
16    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
17    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
18    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
19    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
20    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
21    9999   999999  XXXXXXXXXXXXXXXXXXXXXXXX   9
22 Execution Status: XXXXXXXXXXXXXXXXXXXXXXXX
23
24
```

2.4.3.2 The emulated user must enter, in the appropriate fields of the input/ output screen, the required input data which is divided in two groups and organized as follows:

- Two fields: D_ID and C_ID.

  Comment: The value for *ol_cnt* cannot be entered, but must be determined by the application upon processing of the input data.

- One repeating group of fields: OL_I_ID, OL_SUPPLY_W_ID and OL_QUANTITY. The group is repeated *ol_cnt* times (once per item in the order). The values of these fields are chosen as per Clause 2.4.1.5.

  Comment: In order to maintain a reasonable amount of keyed input, the supply warehouse fields must be filled in for each item, even when the supply warehouse is the home warehouse.

2.4.3.3 The emulated terminal must display, in the appropriate fields of the input/ output screen, all input data and the output data resulting from the execution of the transaction. The display fields are divided in two groups as follows:

- One non-repeating group of fields: W_ID, D_ID, C_ID, O_ID, O_OL_CNT, C_LAST, C_CREDIT, C_DISCOUNT, W_TAX, D_TAX, O_ENTRY_D, *total_amount*, and an optional execution status message other than "Item number is not valid".

- One repeating group of fields: OL_SUPPLY_W_ID, OL_I_ID, I_NAME, OL_QUANTITY, S_QUANTITY, *brand_generic,* I_PRICE, and OL_AMOUNT. The group is repeated O_OL_CNT times (once per item in the order), equal to the computed value of *ol_cnt*.

2.4.3.4 For transactions that are rolled back as a result of an unused item number (1% of all New-Order transactions), the emulated terminal must display in the appropriate fields of the input/ output screen the fields: W_ID, D_ID, C_ID, C_LAST, C_CREDIT, O_ID, and the execution status message "Item number is not valid". Note that no execution status message is required for successfully committed transactions. However, this field may not display "Item number is not valid" if the transaction is successful.

Comment: The number of the rolled back order, O_ID, must be displayed to verify that part of the transaction was processed.

2.4.3.5 The following table summarizes the terminal I/ O requirements for the New-Order transaction:

	Enter	Display	Display	Coordinates
		After rollback	Row/ Column	
Non-repeating		W_ID	W_ID	2/ 12
Group	D_ID	D_ID	D_ID	2/ 29
	C_ID	C_ID	C_ID	3/ 12
		C_LAST	C_LAST	3/ 25
		C_CREDIT	C_CREDIT	3/ 52
		C_DISCOUNT		3/ 64
		W_TAX		4/ 51
		D_TAX		4/ 67
		O_OL_CNT		4/ 42
		O_ID	O_ID	4/ 15
		O_ENTRY_D		2/ 61
		total-amount		22/ 71
			"Item number is not valid"	22/ 19
Repeating Group	OL_SUPPLY_W_ID	OL_SUPPLY_W_ID		7-22/ 3
	OL_I_ID	OL_I_ID		7-22/ 10
		I_NAME		7-22/ 20
	OL_QUANTITY	OL_QUANTITY		7-22/ 45
		S_QUANTITY		7-22/ 51
		brand-generic		7-22/ 58
		I_PRICE		7-22/ 63
		OL_AMOUNT		7-22/ 72

2.4.3.6 For general terminal I/ O requirements, see Clause 2.2.