2.5 The Payment Transaction

The Payment business transaction updates the customer's balance and reflects the payment on the district and warehouse sales statistics. It represents a light-weight, read-write transaction with a high frequency of execution and stringent response time requirements to satisfy on-line users. In addition, this transaction includes non-primary key access to the CUSTOMER table.

2.5.1 Input Data Generation

2.5.1.1 For any given terminal, the home warehouse number (W_ID) is constant over the whole measurement interval.

2.5.1.2 The district number (D_ID) is randomly selected within [1 ..10] from the home warehouse (D_W_ID) = W_ID). The customer is randomly selected 60% of the time by last name (C_W_ID , C_D_ID, C_LAST) and 40% of the time by number (C_W_ID , C_D_ID , C_ID). Independent of the mode of selection, the customer resident warehouse is the home warehouse 85% of the time and is a randomly selected remote warehouse 15% of the time. This can be implemented by generating two random numbers *x* and *y* within [1 .. 100];

- If *x* <= 85 a customer is selected from the selected district number (C_D_ID = D_ID) and the home warehouse number (C_W_ID = W_ID). The customer is paying through his/ her own warehouse.
- If *x* > 85 a customer is selected from a random district number (C_D_ID is randomly selected within [1 .. 10]), and a random remote warehouse number (C_W_ID is randomly selected within the range of active warehouses (see Clause 4.2.2), and C_W_ID ≠W_ID). The customer is paying through a warehouse and a district other than his/ her own.
- If *y* <= 60 a customer last name (C_LAST) is generated according to Clause 4.3.2.3 from a non-uniform random value using the NURand(255,0,999) function. The customer is using his/ her last name and is one of the possibly several customers with that last name.

  Comment: This case illustrates the situation when a customer does not use his/ her unique customer number.
- If *y* > 60 a non-uniform random customer number (C_ID) is selected using the NURand(1023,1,3000) function. The customer is using his/ her customer number.

Comment: If the system is configured for a single warehouse, then all customers are selected from that single home warehouse.

2.5.1.3 The payment amount (H_AMOUNT) is randomly selected within [1.00 .. 5,000.00].

2.5.1.4 The payment date (H_DATE) in generated within the SUT by using the current system date and time.

2.5.1.5 A Payment transaction is said to be **home** if the customer belongs to the warehouse from which the payment is entered (when C_W_ID = W_ID).

2.5.1.6 A Payment transaction is said to be **remote** if the warehouse from which the payment is entered is not the one to which the customer belongs (when C_W_ID does not equal W_ID).

2.5.2 Transaction Profile

2.5.2.1 The Payment transaction enters a customer's payment with a single database transaction and is comprised of:

Case 1, the customer is selected based on customer number:

3 row selections with data retrieval and update,

1 row insertion.

Case 2, the customer is selected based on customer last name:

2 row selections (on average) with data retrieval,

3 row selections with data retrieval and update,

1 row insertion.

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

2.5.2.2 For a given warehouse number (W_ID), district number (D_W_ID , D_ID), customer number (C_W_ID , C_D_ID , C_ ID) or customer last name (C_W_ID , C_D_ID , C_LAST), and payment amount (H_AMOUNT):

- The input data (see Clause 2.5.3.2) are communicated to the SUT.
- A database transaction is started.
- The row in the WAREHOUSE table with matching W_ID is selected. W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE, and W_ZIP are retrieved and W_YTD, the warehouse's year-to-date balance, is increased by H_ AMOUNT.
- The row in the DISTRICT table with matching D_W_ID and D_ID is selected. D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, and D_ZIP are retrieved and D_YTD, the district's year-to-date balance, is increased by H_AMOUNT.
- **Case 1**, the customer is selected based on customer number: the row in the CUSTOMER table with matching C_W_ID, C_D_ID and C_ID is selected. C_FIRST, C_MIDDLE, C_LAST, C_STREET_1, C_STREET_2, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM, C_DISCOUNT, and C_BALANCE are retrieved. C_BALANCE is decreased by H_AMOUNT. C_YTD_PAYMENT is increased by H_AMOUNT. C_PAYMENT_CNT is incremented by 1.

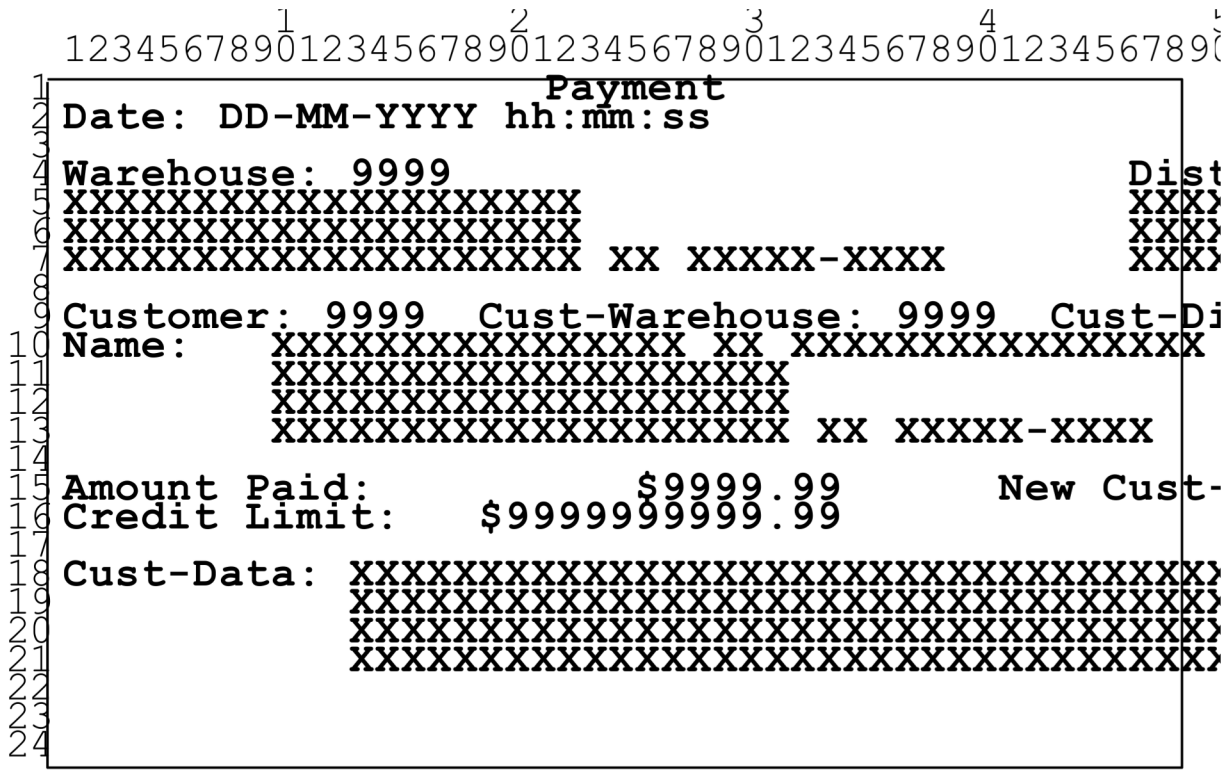
  Case 2, the customer is selected based on customer last name: all rows in the CUSTOMER table with matching C_W_ID, C_D_ID and C_LAST are selected sorted by C_FIRST in ascending order. Let n be the number of rows selected. C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM, C_DISCOUNT, and C_BALANCE are retrieved from the row at position (n/ 2 rounded up to the next integer) in the sorted set of selected rows from the CUSTOMER table. C_BALANCE is decreased by H_AMOUNT. C_YTD_PAYMENT is increased by H_AMOUNT. C_PAYMENT_CNT is incremented by 1.
- If the value of C_CREDIT is equal to "BC", then C_DATA is also retrieved from the selected customer and the following history information: C_ID, C_D_ID, C_W_ID, D_ID, W_ID, and H_AMOUNT, are inserted at the left of the C_DATA field by shifting the existing content of C_DATA to the right by an equal number of bytes and by discarding the bytes that are shifted out of the right side of the C_DATA field. The content of the C_DATA field never exceeds 500 characters. The selected customer is updated with the new C_DATA field. If C_DATA is implemented as two fields (see Clause 1.4.9), they must be treated and operated on as one single field.

Comment: The format used to store the history information must be such that its display on the input/ output screen is in a readable format. (e.g. the W_ID portion of C_DATA must use the same display format as the output field W_ID).

- H_DATA is built by concatenating W_NAME and D_NAME separated by 4 spaces.
- A new row is inserted into the HISTORY table with H_C_ID = C_ID, H_C_D_ID = C_D_ID, H_C_W_ID = C_W_ID, H_D_ID = D_ID, and H_W_ID = W_ID.
- The database transaction is committed.
- The output data (see Clause 2.5.3.3) are communicated to the terminal.

2.5.3 Terminal I/O

2.5.3.1 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. In addition, all address fields (i.e., W_STREET_1, W_STREET_2, W_CITY, W_STATE, and W_ZIP) of the warehouse may display the fixed values for these fields if these values were already retrieved in a previous transaction.



2.5.3.2 The emulated user must enter, in the appropriate fields of the input/ output screen, the required input data which is organized as the distinct fields: D_ID, C_ID or C_LAST, C_D_ID, C_W_ID, and H_AMOUNT.

Comment: In order to maintain a reasonable amount of keyed input, the customer warehouse field must be filled in even when it is the same as the home warehouse.

2.5.3.3 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 following fields are displayed: W_ID, D_ID, C_ID, C_D_ID, C_W_ID, W_STREET_1, W_STREET_2, W_CITY, W_STATE, W_ZIP, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP, C_FIRST, C_MIDDLE, C_LAST, C_STREET_1, C_STREET_2, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM, C_DISCOUNT, C_BALANCE, the first 200 characters of C_DATA (only if C_CREDIT = "BC"), H_AMOUNT, and H_DATE.

2.5.3.4 The following table summarizes the terminal I/ O requirements for the Payment transaction:

	Enter	Display Row/ Column	Coordinates
Non-repeating Group		W_ID	4/ 12
	D_ID	D_ID	4/ 52
	C_ID [1]	C_ID	9/ 11
	C_D_ID	C_D_ID	9/ 54
	C_W_ID	C_W_ID	9/ 33
	H_AMOUNT	H_AMOUNT	15/ 24
		H_DATE	2/ 7
		W_STREET_1	5/ 1
		W_STREET_2	6/ 1
		W_CITY	7/ 1
		W_STATE	7/ 22
		W_ZIP	7/ 25
		D_STREET_1	5/ 42
		D_STREET_2	6/ 42
		D_CITY	7/ 42
		D_STATE	7/ 63
		D_ZIP	7/ 66
		C_FIRST	10/ 9
		C_MIDDLE	10/ 26
	C_LAST [2]	C_LAST	10/ 29
		C_STREET_1	11/ 9
		C_STREET_2	12/ 9
		C_CITY	13/ 9
		C_STATE	13/ 30
		C_ZIP	13/ 33
		C_PHONE	13/ 58
		C_SINCE	10/ 58
		C_CREDIT	11/ 58
		C_CREDIT_LIM	16/ 18
		C_DISCOUNT	12/ 58
		C_BALANCE	15/ 56
		C_DATA [3]	18-21/ 12

[1] Enter only for payment by customer number

[2] Enter only for payment by customer last name

[3] Display the first 200 characters only if C_CREDIT = "BC"

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