

CELLENTA
ORACLE DATABASE
USER MANUAL



Contents

1. TABLES	3
1.1. Subscription Packages	3
1.2. Customer	3
1.3. Balance	3
1.4. Personal Usage	3
1.5. Invoice	4
1.6. Login History	4
1.7. Password Reset Codes	4
1.8. Notification Logs	4
1.9. DML Multi Stage Log	4
2. PROCEDURES	5
2.1. Package	5
2.1.1. get_all_packages	5
2.1.2. get_package	5
2.2. Customer	6
2.2.1. login_customer	6
2.2.2. register_customer	7
2.2.3. Insert Password Reset Code	7
2.2.4. verify_password_reset_code	8
2.2.5. get_customer	9
2.2.6. change_customer_password	9
2.2.7. get_customer_by_id	10
2.2.8. get_msisdns_haspackage	10
2.3. Balance	11
2.3.1. get_balance	11
2.3.2. update_balance_all	11
2.3.3. add_new_balance	12
2.3.4. insert_personal_usage	12
2.3.5. get_customer_invoices	13
2.4. Logging	14
2.4.1. log_notification	14
2.4.2. get_last_30_logs_by_user	15
2.4.3. get_error_logs_by_user	15
3. USERS AND AUTHORIZATIONS	17
3.1. AOM Users	17
3.2. ABMF Users	17
3.3. CGF Users	18
3.4. NS Users	18
3.5. Test Users	18

1. TABLES



1.1. Subscription Packages

Stores information about the available subscription plans, including package names, included data/minutes/SMS amounts, and pricing details.

1.2. Customer

Contains customer-related information such as personal details, contact information, and identifiers used in the system.

1.3. Balance

Keeps track of users' remaining quota, including data, minutes, and SMS balances, along with the corresponding validity periods (start and end dates).

1.4. Personal Usage

Logs individual usage events such as call durations, number of sent SMS messages, and data consumption, providing a detailed breakdown of how each user utilizes their plan.

1.5.Invoice

Stores historical billing data, including the amount charged, billing period, and payment status for each customer.

1.6.Login History

Records user login attempts, including timestamps, user identifiers, and source IPs, primarily for audit and security monitoring purposes.

1.7.Password Reset Codes

Holds temporary reset tokens created when users request a password change, along with expiration times and related IPs.

1.8.Notification Logs

Captures logs of alert notifications (e.g., low balance warnings) sent via email , ensuring traceability of outbound communication.

1.9.DML Multi Stage Log

Logs all DML (Data Manipulation Language) operations—such as INSERT, UPDATE, SELECT—performed by database users in three distinct stages: Started, Processing, and Completed (or Error if failed), along with detailed error messages when applicable.

2. PROCEDURES

Status Codes Will be added here

2.1. *Package*

2.1.1. **get_all_packages**

Retrieves all available subscription packages stored in the SUBSCRIPTION_PACKAGES table.

Status Codes:

- 200: Operation completed successfully.
- 500: General error occurred during execution.

Inputs:

- *None*

Outputs:

- o_packages (OUT, SYS_REFCURSOR): Returns a list of all subscription packages with the following fields:
 - PACKAGE_ID: Unique identifier of the package (NUMBER)
 - PACKAGE_NAME: Name of the subscription package (VARCHAR2)
 - PRICE: Price of the package (NUMBER)
 - AMOUNT_MINUTES: Included voice minutes (NUMBER)
 - AMOUNT_DATA: Included mobile data (NUMBER)
 - AMOUNT_SMS: Included SMS count (NUMBER)
 - PERIOD: Duration of the package in days or months (NUMBER)
- o_status_code (OUT, NUMBER): Indicates the status of the procedure

2.1.2. **get_package**

Retrieves the subscription package details for the given PACKAGE_ID.

Status Codes:

- 200: Package found and returned successfully.
- 404: No package found for the given ID.
- 500: General error occurred during execution.

Inputs:

- p_package_id (IN, NUMBER): The ID of the subscription package to be retrieved.

Outputs:

- o_package (OUT, SYS_REFCURSOR): Returns the package information if found, with the following fields:
 - PACKAGE_ID: Unique identifier of the package (NUMBER)
 - PACKAGE_NAME: Name of the subscription package (VARCHAR2)
 - PRICE: Price of the package (NUMBER)
 - AMOUNT_MINUTES: Included voice minutes (NUMBER)
 - AMOUNT_DATA: Included mobile data (NUMBER)
 - AMOUNT_SMS: Included SMS count (NUMBER)
 - PERIOD: Duration of the package in days or months (NUMBER)
- o_status_code (OUT, NUMBER): Indicates the status of the procedure execution.

2.2. Customer

2.2.1. login_customer

Handles customer login by verifying the provided MSISDN and password.

Status Codes:

- 200: Login successful.
- 401: Incorrect password.
- 404: Customer not found.
- 500: General error occurred during execution.

Inputs:

- p_msisdn (IN, VARCHAR2): MSISDN (phone number) of the customer attempting to log in.
- p_password (IN, VARCHAR2): Password entered by the customer.
- p_device_type (IN, VARCHAR2, default: 'notDefined'): Type of device used for login.
- p_ip_address (IN, VARCHAR2, default: 'notDefined'): IP address of the customer's device.

Outputs:

- o_customer (OUT, SYS_REFCURSOR): If login is successful, returns the following customer information:
 - CUSTOMER_ID: Unique identifier of the customer (NUMBER)
 - MSISDN: Customer's phone number (VARCHAR2)
 - NAME: First name of the customer (VARCHAR2)
 - SURNAME: Last name of the customer (VARCHAR2)
 - EMAIL: Email address (VARCHAR2)
 - CREATED_AT: Account creation timestamp (TIMESTAMP)
- o_status_code (OUT, NUMBER): Indicates the result of the login attempt.

2.2.2. register_customer

Registers a new customer by saving their personal and contact information into the system.

Status Codes:

- 200: Registration successful.
- 500: General error occurred during execution.

Inputs:

- p_msisdn (IN, VARCHAR2): Customer's phone number (MSISDN).
- p_name (IN, VARCHAR2): First name of the customer.
- p_surname (IN, VARCHAR2): Last name of the customer.
- p_email (IN, VARCHAR2): Email address of the customer.
- p_password (IN, VARCHAR2): Password chosen by the customer.

Outputs:

- o_customer_id (OUT, NUMBER): The newly created customer's unique ID.
- o_result (OUT, NUMBER): Status code indicating the result of the registration.

2.2.3. Insert Password Reset Code

Generates and stores a password reset code for the user based on their email address.

Status Codes:

- 200: Reset code generated and stored successfully.
- 500: General error occurred during execution.

Inputs:

- p_email (IN, VARCHAR2): Email address of the user requesting a password reset.
- p_ip_address (IN, VARCHAR2, default: NULL): IP address of the device making the request.
- p_device_type (IN, VARCHAR2, default: NULL): Type of device used for the request.

Outputs:

- o_validity_period (OUT, NUMBER): Validity duration (in minutes or seconds) of the generated reset code.
- o_reset_code (OUT, VARCHAR2): The actual reset code generated for password recovery.
- o_status_code (OUT, NUMBER): Indicates the result of the operation.

2.2.4. verify_password_reset_code

Validates a submitted password reset code for a given email address.

Status Codes:

- 200: Reset code is valid and unused.
- 404: Reset code not found for the provided email.
- 409: Reset code has already been used.
- 410: Reset code has expired.
- 500: General error occurred during execution.

Inputs:

- p_email (IN, VARCHAR2): Email address associated with the reset code.
- p_reset_code (IN, VARCHAR2): Password reset code submitted by the user.

Outputs:

- o_status_code (OUT, NUMBER): Indicates the result of the verification attempt.

2.2.5. get_customer

Retrieves customer information based on the provided MSISDN (phone number).

Status Codes:

- 200: Customer found and returned successfully.
- 404: No customer found with the given MSISDN.
- 500: General error occurred during execution.

Inputs:

- p_msisdn (IN, VARCHAR2): The MSISDN (phone number) of the customer to retrieve.

Outputs:

- o_customer (OUT, SYS_REFCURSOR): Returns customer details including:
 - CUSTOMER_ID: Unique customer identifier (NUMBER)
 - MSISDN: Customer's phone number (VARCHAR2)
 - NAME: First name (VARCHAR2)
 - SURNAME: Last name (VARCHAR2)
 - EMAIL: Email address (VARCHAR2)
 - PASSWORD: Hashed password (VARCHAR2)
 - CREATED_AT: Account creation timestamp (TIMESTAMP)
- o_status_code (OUT, NUMBER): Indicates the status of the procedure execution.

2.2.6. change_customer_password

Updates the password of a customer based on their email address.

Status Codes:

- 200: Password updated successfully.
- 404: No customer found with the given email.
- 500: General error occurred during execution.

Inputs:

- p_email (IN, VARCHAR2): Email address of the customer requesting the password change.
- p_new_password (IN, VARCHAR2): New password to be set for the customer.

Outputs:

- o_status_code (OUT, NUMBER): Indicates the result of the operation

2.2.7. get_customer_by_id

Retrieves customer information based on the provided customer ID.

Status Codes:

- 200: Customer found and returned successfully.
- 404: No customer found with the given customer ID.
- 500: General error occurred during execution.

Inputs:

- p_customer_id (IN, VARCHAR2): The unique identifier of the customer to retrieve.

Outputs:

- o_customer (OUT, SYS_REFCURSOR): Returns customer details including:
 - CUSTOMER_ID: Unique customer identifier (VARCHAR2)
 - MSISDN: Customer's phone number (VARCHAR2)
 - NAME: First name (VARCHAR2)
 - SURNAME: Last name (VARCHAR2)
 - EMAIL: Email address (VARCHAR2)
 - PASSWORD: Hashed password (VARCHAR2)
 - CREATED_AT: Account creation timestamp (TIMESTAMP)
- o_status_code (OUT, NUMBER): Indicates the status of the procedure execution.

2.2.8. get_msisdns_haspackage

Retrieves all MSISDNs (phone numbers) of customers who have an associated balance record, indicating an active subscription package.

Status Codes:

- 200: Operation completed successfully.
- 500: General error occurred during execution.

Inputs:

- *None*

Outputs:

- o_msisdns (OUT, SYS_REFCURSOR): Returns a list of MSISDNs belonging to customers with active packages.
- o_status_code (OUT, NUMBER): Indicates the status of the procedure execution.

2.3. Balance

2.3.1. get_balance

Retrieves the remaining balance details for a customer identified by their MSISDN.

Status Codes:

- 200: Balance information retrieved successfully.
- 404: No balance record found for the given MSISDN.
- 500: General error occurred during execution.

Inputs:

- p_msisdn (IN, VARCHAR2): The MSISDN of the customer whose balance is requested.

Outputs:

- o_balance (OUT, SYS_REFCURSOR): Returns balance details including:
 - CUSTOMER_ID: Unique customer identifier (NUMBER)
 - PACKAGE_ID: Associated subscription package ID (NUMBER)
 - REMAINING_MINUTES: Remaining voice minutes (NUMBER)
 - REMAINING_SMS: Remaining SMS count (NUMBER)
 - REMAINING_DATA: Remaining data volume (NUMBER)
 - START_DATE: Balance validity start date (DATE or TIMESTAMP)
 - END_DATE: Balance validity end date (DATE or TIMESTAMP)
- o_status_code (OUT, NUMBER): Indicates the status of the procedure execution.

2.3.2. update_balance_all

Updates the remaining balance values (minutes, SMS, data) for a customer identified by their MSISDN.

Status Codes:

- 200: Balance updated successfully.
- 404: No balance record found for the given MSISDN.
- 500: General error occurred during execution.

Inputs:

- p_msisdn (IN, VARCHAR2): The MSISDN of the customer whose balance will be updated.
- p_new_minutes (IN, NUMBER): New value for remaining voice minutes.
- p_new_sms (IN, NUMBER): New value for remaining SMS count.
- p_new_data (IN, NUMBER): New value for remaining data volume.

Outputs:

- o_status_code (OUT, NUMBER): Indicates the result of the update operation.

2.3.3. add_new_balance

Assigns a new balance record to a customer based on their customer ID and the selected subscription package.

Status Codes:

- 200: New balance record added successfully.
- 500: General error occurred during execution.

Inputs:

- p_customer_id (IN, NUMBER): Unique identifier of the customer.
- p_package_id (IN, NUMBER): Identifier of the subscription package to assign.

Outputs:

- o_status_code (OUT, VARCHAR): Indicates the result of the operation.

2.3.4. insert_personal_usage

Records personal usage details such as call minutes, SMS sent, or data consumption for a customer.

Important Note:

- The p_usage_type parameter accepts values: minutes, sms, or data.

- If p_usage_type is minutes or sms, the p_receiver_msisdn must be provided.
- Data usage is stored in megabytes (MB).

Status Codes:

- 200: Usage recorded successfully.
- 404: Related customer or record not found.
- 500: General error occurred during execution.

Inputs:

- p_giver_msisdn (IN, VARCHAR2): MSISDN of the user who generated the usage (caller or sender).
- p_receiver_msisdn (IN, VARCHAR2, default: NULL): MSISDN of the receiver (required for minutes and SMS usage).
- p_usage_date (IN, TIMESTAMP): Date and time of the usage event.
- p_usage_type (IN, VARCHAR2): Type of usage (minutes, sms, or data).
- p_usage_duration (IN, NUMBER): Duration or amount of usage (minutes, count of SMS, or data in MB).

Outputs:

- o_status_code (OUT, NUMBER): Indicates the result of the insertion.

2.3.5. get_customer_invoices

Retrieves invoices related to a given MSISDN, including both historical invoices and the current active package invoice with remaining days.

Status Codes:

- 200: Invoices retrieved successfully.
- 404: No invoices found for the given MSISDN.
- 500: General error occurred during execution.

Inputs:

- p_msisdn (IN, VARCHAR2): The MSISDN (phone number) of the customer whose invoices are requested.

Outputs:

- o_cursor (OUT, SYS_REFCURSOR): Returns invoice records with the following fields:

- INVOICE_ID: Invoice identifier (NUMBER or NULL for active package without invoice)
- CUSTOMER_ID: Customer identifier (NUMBER)
- PACKAGE_ID: Associated package identifier (NUMBER)
- START_DATE: Invoice or package start date (VARCHAR2, formatted as DD/MM/YY)
- END_DATE: Invoice or package end date (VARCHAR2, formatted as DD/MM/YY)
- PRICE: Price charged (NUMBER)
- PAYMENT_STATUS: Payment status (e.g., PAID, UNPAID)
- IS_ACTIVE: 'Y' if the package is currently active, 'N' otherwise
- DAYS_LEFT: Number of days remaining for the active package (NUMBER), NULL otherwise
- o_status_code (OUT, NUMBER): Indicates the result of the procedure execution.

2.4. Logging

2.4.1. log_notification

Logs notifications sent to users regarding their remaining package usage.

Status Codes:

- 200: Notification logged successfully.
- 404: User or related record not found.
- 500: General error occurred during execution.

Inputs:

- p_email (IN, VARCHAR2): Email address to which the notification was sent.
- p_notification_type (IN, VARCHAR2): Type of notification sent (e.g., low balance alert, usage warning).

Outputs:

- o_status_code (OUT, VARCHAR): Indicates the result of the logging operation.

2.4.2. get_last_30_logs_by_user

Returns the last 30 DML operation logs for the currently connected Oracle user.

Status Codes:

- 200: Logs retrieved successfully.
- 500: General error occurred during execution.

Inputs:

- *None explicitly required.* The procedure internally uses the session's Oracle user (v_username).

Outputs:

- o_logs (OUT, SYS_REFCURSOR): Returns up to 30 recent DML log entries for the user, including details such as:
 - Operation type (SELECT, INSERT, etc.)
 - Target table
 - Stage of operation (STARTED, PROCESSING, COMPLETED, ERROR)
 - Operation time (TIMESTAMP)
 - Error message, if any
- o_status_code (OUT, NUMBER): Indicates the result of the log retrieval operation.

2.4.3. get_error_logs_by_user

Retrieves all error logs related to DML operations performed by the currently connected Oracle user.

Status Codes:

- 200: Error logs retrieved successfully.
- 500: General error occurred during execution.

Inputs:

- *None explicitly required.* The procedure automatically uses the session's Oracle username (v_username).

Outputs:

- o_logs (OUT, SYS_REFCURSOR): Returns DML log entries where the operation status is 'ERROR', including:
 - USER_NAME: Oracle user who executed the operation

- TABLE_NAME: Affected table
 - OPERATION: Operation type (e.g., SELECT, INSERT)
 - STAGE: Operation stage (e.g., STARTED, PROCESSING, ERROR)
 - ERROR_MESSAGE: Description of the error encountered
 - OPERATION_TIME: Timestamp of the operation
- o_status_code (OUT, NUMBER): Indicates the result of the procedure execution.

3. USERS AND AUTHORIZATIONS

The following two procedures are available for **all users**:

- get_last_30_logs_by_user
- get_error_logs_by_user

3.1.AOM Users

This user is responsible for interacting with the UI layer to fetch persistent data such as customer details and subscription packages. It also manages communication with the mail service for password reset operations (e.g., sending verification codes). Additionally, AOM_USERS pushes user-related data into Hazelcast to simulate backend traffic where needed.

Authorized Procedures:

- login_customer
- register_customer
- get_package
- get_all_packages
- add_new_balance
- get_customer_by_id
- get_msisdns_haspackage
- get_balance
- insert_password_reset_code
- verify_password_reset_code
- change_customer_password
- get_customer_invoices

3.2.ABMF Users

Responsible for updating balance information on Oracle side in a delayed/asynchronous manner.

Authorized Procedures:

- get_balance
- update_balance_all

3.3.CGF Users

Responsible for processing package usage records asynchronously or with delay.

Authorized Procedures:

- insert_personal_usage

3.4.NS Users

Responsible for retrieving the email address linked to a given phone number and logging any sent notifications.

Authorized Procedures:

- get_customer
- log_notification

3.5.Test Users

Responsible for testing the entire system. Has access to all procedures.