IT Mentor, 3rd Floor, Pranavam arcade, Above Chinese wok, Outer Ring Rd, Hemanth Nagar, Marathahalli, Bengaluru 560 037

Oracle Apps

OREDR TO CASH CYCLE INTERVIEW QUESTIONS

1) **Explain me the o2c cycle along with tables**- covered in the notes, sent separate doc for this
2) **What will happen at interface trip stop?** - covered in the notes
3) **What is auto ship**- covered in the notes
4) **What is the api for shipping**

   wsh_deliveries_pub. Delivery_Action
5) **What are the mandatory parameters for api**

```
api_version_number     IN   NUMBER,
p_init_msg_list        IN   VARCHAR2,
x_return_status        OUT NOCOPY   VARCHAR2,
x_msg_count            OUT NOCOPY   NUMBER,
x_msg_data             OUT NOCOPY   VARCHAR2,
```

6) **purpose of Workflow Background Process**

   Workflow Background Process picks and executes all eligible workflow items related to Oracle Modules like OM,AP,PO,WSH,OKL,ASN,WIP, etc
   Workflow Background Process is a concurrent program which is run for processing **deferred** activities, **timed out** activities, and **stuck** processes using the parameters specified Hence if ItemType parameter is not specified the workflow background process runs will first check for all eligible workflow activities and then run for all eligible workflows this will result in poor performance. It is preferable to execute the Workflow Background Process with specified ItemType.
7) **workflow background parameters**

Parameters for "Workflow background process" concurrent program

a. Item Type: If you want to restrict workflow engine to a specific Item Type, specify the same here. Else, the engine will process all/any deferred activity regardless of the Item Type.
b. Minimum Threshold: If you want to restrict this engine to activities with specific minimum cost, define it here. Otherwise the Workflow engine will process any deferred activity regardless of cost.
c. Maximum Threshold: If you want to restrict this engine to activity with specific maximum cost, define it here. Otherwise the workflow engine will process any deferred activities regardless of cost.
d. Process Deferred: Specify whether the workflow engine should run deferred. Set it to Yes or No. If this set to yes, the Workflow Background Process will run the workflow activities as a deferred process.
e. Process Time out: Specify whether the workflow engine should check for activities that have been timed out. Set it to Yes or No.

8) how to submit workflow back ground process from back end

```
BEGIN
 wf_engine.background (itemtype=>NULL ,
                       min threshold=>NULL ,
                       maxthreshold=>NULL ,
                       process_deferred=>TRUE ,
                       process_timeout=>FALSE ,
                       process_stuck=>FALSE);
END;
```

9) who will run workflow back ground process in the real time

no one will run, it is an automated program i.e. scheduled program.

10) at what point of time sales order line will be closed

after invoice created

11) while creating invoice how many tables affected

ra_customer_Trx_all, ra_customer_Trx_lines_all, ar_payment_schedules_all

12). how many orders line level statuses u know in o2c

Shipped, waiting shipping, closed, entered, production open, supply eligible, EXTERNAL_REQ_REQUESTED, EXTERNAL_REQ_open, Awaiting receipt

13). **in how many ways we can create sales order order import concurrent program manual**

oe_order_pub.process_order

```
Process_Order
(   p_org_id                    IN  NUMBER  := NULL  --MOAC
,   p_operating_unit            IN  VARCHAR2 := NULL -- MOAC
,   p_api_version_number        IN  NUMBER
,   p_init_msg_list             IN  VARCHAR2 := FND_API.G_FALSE
```

```
, p_return_values          IN  VARCHAR2 := FND_API.G_FALSE
, p_action_commit          IN  VARCHAR2 := FND_API.G_FALSE
, x_return_status         OUT NOCOPY VARCHAR2
, x_msg_count             OUT NOCOPY NUMBER
, x_msg_data              OUT NOCOPY
```

14). order import concurrent program

navigation – order management responsibility
order returns—import orders—order import request
operating unit, order source, order reference, validate only

15). how to display sales rep

```
select jrd.resource_name sales_rep

from oe_order_headers_all ooh,

    jtf_rs_salesreps jrs,

    jtf_rs_defresources_v jrd

    where ooh.SALESREP_ID=jrs.salesrep_id

    and jrs.resource_id=jrd.resource_id

    and ooh.order_number='69479'
```

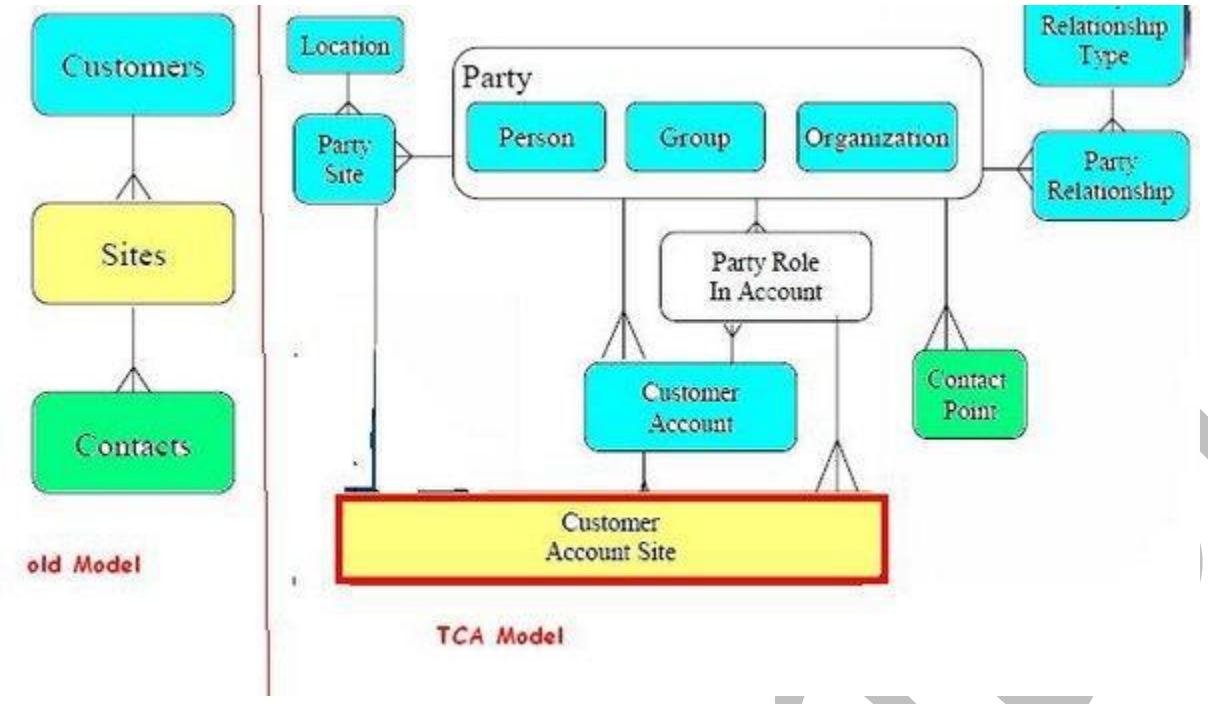
16) how to derive "bill to”,” ship to" location?

17) TCA- Trading community Architecture.

TCA is like central repository for the entire E-Business Suite to store information relating to all members of a trading community versus separate tables for each member-Prospects, Customers, Contacts, Employees, Partners, Distributors, Suppliers, Banks, etc.

Record complex business relationships between Trading Community entities (including 3rd party relationships).

Support all business models, industries, and geographies. architecture—trading community architecture



party-- hz_parties

account-- hz_cust_accounts

party site-- HZ_PARTY_SITES

party site uses - HZ_PARTY_SITE_USES

account site-- HZ_CUST_ACCT_SITES

account site uses-- HZ_CUST_SITE_USES

location-- HZ_LOCATIONS

18) sales order holds and its table and releases table

Actions- apply holds

oe_holds.HOLDS_API

oe_hold_definitions

19) how to release holds

Release table

oe_hold_releases

19) what are the concurrent programs after pick release

Pick Slip Report

Shipping Exceptions Report

20) what are the item attributes to enable for item in o2c cycle

Customer ordered, customer orders enabled, shippable, oe transactable.

21) back order-

1. Order on Hold
2. Inventory Period NOT open
3. No enough on-hand quantity
4. No enough quantity to reserve/transact
5. No on-hand quantity in required sub-inventory
6. The Lot from which items are selected is inactive/expired
7. Wrong Item reservation (even inventory have enough quantity)
8. Inventory reserved for other sales orders
9. Inventory picked-up by other sales orders
10. Previously done return to stock not properly performed
11. Move order is in pending state

22)till what point of time we can cancel sales order line

only order entry, booking
if we try to do at pick release you will get below issue
You are not allowed to cancel Order Line because:
Line has been pick confirmed/staged.

23) how to crate invoice automatically

auto invoice import-- to create invoice automatically

24) link between om and ar header level?

ra_customer_Trx_all.interface_header_attribute1 = TO_CHAR (oe_order_headers_all.order_number)

25) link between om line level and ar line level

AND ra_customer_Trx_lines_all.interface_line_attribute6 = TO_CHAR(oe_order_lines_all.line_id)

26) customer outstanding amount /balance amount

ar_payment_schedules_all

due_date, amount_due_original, amount_due_remaining—main columns

27) types of sales orders

Ship only, return only, regular, standard, web, internal, mixed

28) types of AR invoice Types

Invoice, credit memo, debit memo, charge back, guaranty, deposit,

29) how to display open invoices

```
select * from ra_customer_trx_all where open_flag='N'
```

30) derive credit memo

```
SELECT rct.trx_number invoice_number, credit.trx_number credi_memo
  FROM ra_customer_trx_all rct, ra_customer_trx_all credit
 WHERE 1 = 1
   AND rct.customer_trx_id = credit.previous_customer_trx_id
   AND rct.trx_number = '12086'
```

31) how to create credit memo

1) create ar invoice
2) create accounting.
3) tools- credit
enter negative amount
4) create accounting
5) query

```
/* Formatted on 5/16/2018 11:41:35 AM (QP5 v5.139.911.3011) */
SELECT rcta.trx_number, credit.trx_number creditmemo
  FROM ra_customer_Trx_all rcta, ra_customer_trx_all credit
 WHERE rcta.customer_Trx_id = credit.previous_customer_Trx_id
      --AND rcta.trx_number = '500875'
      and rcta.org_id=204
```

32) ar receipt tables

AR_CASH_RECEIPTS_ALL

AR_CASH_RECEIPT_HISTORY_ALL

AR_RECEIVABLE_APPLICATIONS_ALL

33) auto lock box in account receivables

34) sub ledger accounting tables

xla_ae_headers h ,xla_ae_lines l ,xla_events e ,xla_transaction_entities te

xla_distribution_links

35) create accounting.

1. **Create Draft Accounting:**

   Draft will create journal entries, which are not final, which means they are not ready to be transferred to GL.
   You can see the accounting in xla_ae_headers and xla_ae_lines. But the xla_ae_headers.accounting_entry_status_code is D and also the xla_events.process_status_code is D and event_status_code is U.
   You can run create accounting on this transaction again and again, which will delete the old journal

entries and create new ones.
You can't transfer these journal entries to GL.

2.Create Final Accounting:

Final will create journal entries, which can be transferred to GL.
You can see the accounting in xla_ae_headers and xla_ae_lines. But the xla_ae_headers.accounting_entry_status_code is F and also the xla_events.process_status_code is P and event_status_code is P.
Once it is finally accounted you can't run create accounting on the particular transaction (specifically on that event).
You can transfer them to GL using Transfer Journal Entries to GL program.

3. Create Final Accounting Post To GL:

Final Post will create journal entries in final mode, transfer them to GL and post them.
You can see the accounting in xla_ae_headers and xla_ae_lines. But the xla_ae_headers.accounting_entry_status_code is F and also the xla_events.process_status_code is P and event_status_code is P.
Once it is finally accounted you can't run create accounting on the particular transaction (specifically on that event).
It will transfer the journal entries to GL using Journal Import and you can find the data in gl_je_headers and gl_je_lines.
Now the xla_ae_headers.transfer_status_code is Y.
And also it will post to gl_balances (gl_je_headers.status is P).

36) **purpose of journal import-** to create journal entries in GL

37) gl tables

GL_JE_BATCHES: Stores journal entry batches. Each row includes the batch name, description, status, running total debits and credits, and other information.
GL_JE_HEADERS: Stores journal entries. There is a one-to-many relationship between journal entry batches and journal entries. Each row in this table includes the associated batch ID, the journal entry name and description, and other information about the journal entry.
GL_JE_LINES: Stores the journal entry lines that you enter in the Enter Journals form. There is a one-to-many relationship between journal entries and journal entry lines. Each row in this table stores the associated journal entry header ID, the line number, the associated code combination ID, and the debits or credits associated with the journal line.

GL_JE_SOURCES: Stores journal entry source names and descriptions. Each journal entry in your Oracle General Ledger application is assigned a source name to indicate how it was created. This table corresponds to the Journal Sources form.
GL_JE_CATEGORIES: Stores journal entry categories. Each row includes the category name and description.
GL_DAILY_RATES: Stores the daily conversion rates for foreign currency transactions. It replaces the GL_DAILY_CONVERSION_RATES table. It stores the rate to use when converting between two currencies for a given conversion date and conversion type.
GL_DAILY_BALANCES: Stores daily aggregate balances for detail and summary balance sheet accounts in sets of books with average balances enabled.
GL_BUDGET_TYPES: Stores information about budget types. Oracle General Ledger supports only one budget type, 'STANDARD'. Therefore, this table always contains only one row.

GL_INTERFACE: It is used to import journal entry batches through Journal Import. You insert rows in this table and then use the Import Journals window to create journal batches.
GL_INTERFACE_CONTROL: It is used to control Journal Import execution. Whenever you start Journal Import from the Import Journals form, a row is inserted into this table for each source and group id that you specified. When Journal Import completes, it deletes these rows from the table.
GL_BUDGET_INTERFACE: It is used to upload budget data into your Oracle General Ledger application from a spreadsheet program or other external source. Each row includes one fiscal year's worth of budget amounts for an account.

38) gl period types

Weekly, monthly, quarterly, half yearly, yearly

39) difference between primary ledger and secondary ledger

A primary ledger is the main ledger that records an organization's financial transactions, while a secondary ledger can be created to record transactions in a different currency or multi reporting currency or accounting method to meet specific accounting requirements or reporting needs.

40) what is chart of accounts

In Oracle Apps, a chart of accounts is a structured list of all the accounts used to record financial transactions within an organization

The chart of accounts is structured hierarchically, with each account assigned a unique account code or number.

The chart of accounts is integrated with other modules in Oracle Apps, such as General Ledger, Accounts Receivable, Accounts Payable, and Fixed Assets.

The chart of accounts is used to generate financial reports, such as balance sheets, income statements, and cash flow statements. It provides the basis for financial analysis and decision-making.

41**) kff(key flex field) in gl**- Accounting key flex fields

42) **Journal types in oracle apps**

In Oracle Apps, journal types are used to classify accounting entries based on their purpose or source. Journal types are predefined and are used to ensure that accounting entries are recorded and processed consistently across the organization. Here are some common journal types in Oracle Apps:

Standard Journal: This is the most common journal type in Oracle Apps. It is used to record day-to-day transactions, such as sales, purchases, and expenses.

Adjustment Journal: This journal type is used to make adjustments to previously recorded transactions, such as correcting errors or allocating costs.

Reversal Journal: This journal type is used to reverse a previously recorded transaction. It is often used to correct errors or to cancel a transaction that has been recorded in error.

Mass Allocation Journal: This journal type is used to allocate costs or revenues to multiple accounts based on a predefined formula.

Budget Journal: This journal type is used to record budget transactions, such as creating, revising, or transferring budgets.

Intercompany Journal: This journal type is used to record transactions between companies within the same organization. It is used to eliminate intercompany transactions and to ensure that financial statements reflect the true financial position of the organization.

In summary, journal types in Oracle Apps are used to classify accounting entries based on their purpose or source. They ensure that accounting entries are recorded and processed consistently across the organization and provide a framework for financial reporting and analysis.

43) gl budget types in oracle apps

In Oracle Apps, GL (General Ledger) budget types are used to classify and manage different types of budgets for an organization. Budgets are a critical component of financial planning and control, and GL

budget types provide a flexible framework for managing budgets. Here are some common GL budget types in Oracle Apps:

Original Budget: This is the initial budget that is created at the beginning of a fiscal year or period.

Current Budget: This is the budget that is updated periodically throughout the fiscal year or period.

Supplemental Budget: This budget type is used to allocate additional funds to specific departments or projects. It is often used for unexpected expenses or changes in business strategy.

Rollup Budget: This budget type consolidates budgets from multiple departments or business units. It provides a top-down view of the organization's financial performance and is used for high-level financial planning and analysis.

Zero-Based Budget: This budget type requires all expenses to be justified for each fiscal year or period. It is used to control costs and ensure that resources are allocated efficiently.

Forecast Budget: This budget type is used to predict financial performance for a future period. It is often used for long-term financial planning and strategy development.

44) **how to know gl period statuses** i.e. table—Gl_period_statuses

45**) how to display opening balance and closing balance in gl**

```
SUM(NVL(gb.begin_balance_dr,0)-NVL(gb.begin_balance_cr,0)) Opening balance,
SUM(NVL(gb.begin_balance_dr,0)-NVL(gb.begin_balance_cr,0) +
(NVL(gb.period_net_Dr,0) - NVL(gb.period_net_cr,0))) Closing Bal
FROM gl_balances gb,
gl_code_combinations_kfv gcc
```

46) **cash receipt api name in receivables**

AR_RECEIPT_API_PUB

47) **shipping api in order management**

Wsh_delivery_pub

48) what are processing constraints in order management

In Oracle Apps Order Management, processing constraints can be used to control various aspects of order processing, including pricing, shipping, and scheduling. Here are some examples of processing constraints that can be set up in Order Management:

Pricing Constraints: These constraints can be used to restrict the pricing options available to a user based on certain criteria, such as customer, product, or order type. For example, a pricing constraint could prevent a user from applying a certain discount to an order if the customer has not met certain criteria.

Shipping Constraints: These constraints can be used to restrict the shipping options available to a user based on certain criteria, such as location, weight, or order value. For example, a shipping constraint could prevent a user from shipping to a certain location if it is not allowed by the shipping carrier.

Scheduling Constraints: These constraints can be used to restrict the scheduling options available to a user based on certain criteria, such as product availability or order priority. For example, a scheduling constraint could prevent a user from scheduling an order for a certain product if it is not available in inventory.

Approval Constraints: These constraints can be used to restrict the approval options available to a user based on certain criteria, such as order value or customer credit limit. For example, an approval constraint could prevent a user from approving an order if it exceeds a certain dollar amount.

What are defaulting rules in order management?

In Oracle Apps Order Management, defaulting rules are used to automatically populate certain fields with default values based on predefined rules. These rules can be set up at various levels, such as organization, customer, and product, and can be used to streamline the order entry process and ensure data accuracy.

Here are some examples of defaulting rules that can be set up in Order Management:

Payment Terms: This defaulting rule automatically populates the payment terms field on an order based on the customer's payment terms profile or the default payment terms defined for the organization.

Ship-To Location: This defaulting rule automatically populates the ship-to location field on an order based on the customer's default ship-to address or the default ship-to location defined for the organization.

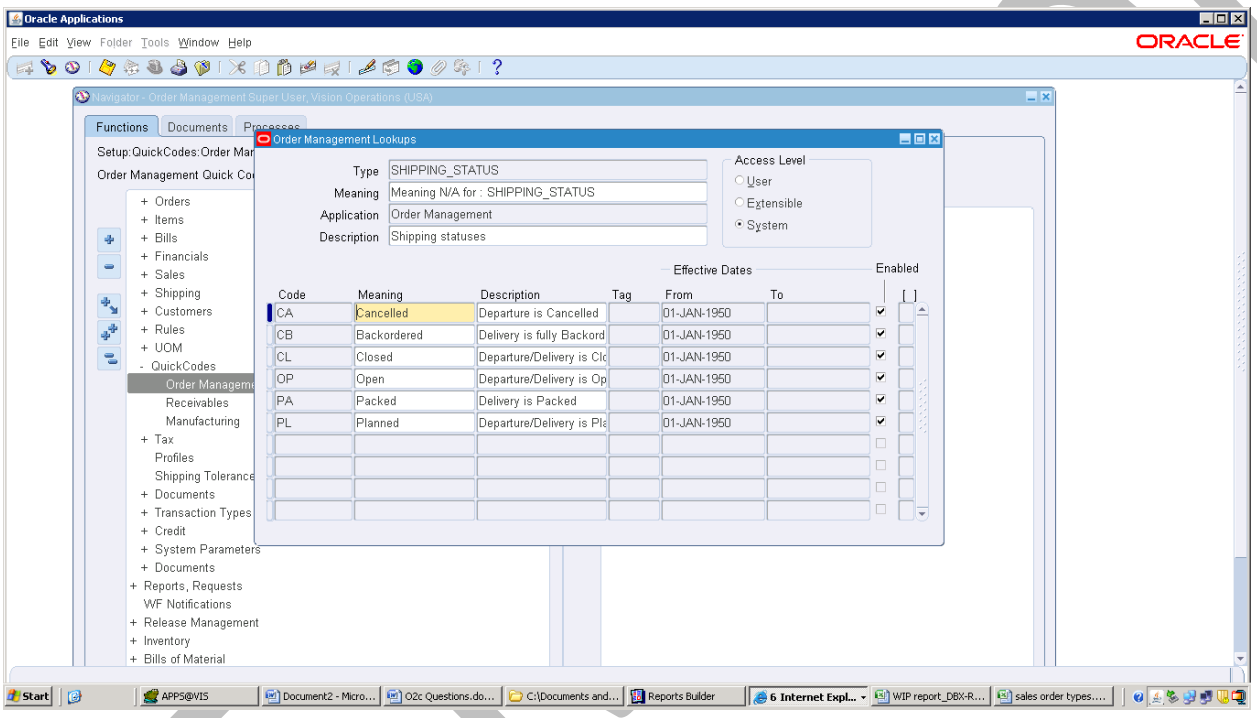
Pricing: This defaulting rule automatically applies a default pricing strategy, such as list price or contract price, to an order based on the customer's pricing profile or the default pricing strategy defined for the product or organization.

Shipping Method: This defaulting rule automatically populates the shipping method field on an order based on the customer's default shipping method or the default shipping method defined for the organization.

Salesperson: This defaulting rule automatically populates the salesperson field on an order based on the customer's default salesperson or the default salesperson defined for the organization.

49) what are quick codes in order management

Setup-quick codes- order management



50) ar to sla to gl link query—covered in notes

51) accounting entries in o2c

Invoice

Receivable – dr

Revenue – cr

Receipts

Receivable – cr

Cash clearance – dr

52) what is move order and its tables in o2c

Mtl_txn_request_headers

Mtl_trx_request_lines

53) how to join sales order table and customer- covered in notes

54) how to join customer and invoice table

hz_cust_accounts .cust_account_id= ra_customer_trx_all.bill_to_customer_id

55) how to use customer as supplier

This feature is commonly used when a company buys goods or services from a customer, or when a customer is also a vendor.

There are several scenarios in which you might use a customer as a supplier in Oracle Apps:

Consignment Sales: If you are in the business of selling consignment items, a customer might also act as a supplier by providing goods that you sell on their behalf. In this case, you would purchase the items from the customer/supplier and sell them to your customers.

Customer Returns: If a customer returns goods to you, they might also act as a supplier by providing replacement items or credits that you can use to purchase other goods from them.

Joint Ventures: If you are in a joint venture with a customer, they might also act as a supplier by providing goods or services to your organization. In this case, you would need to purchase these items from them in order to complete the joint venture project.

56) gl translation vs revaluation—covered in previous question

57) link between om header level and ar line level

to_char(oe_order_headers_all.order_number) =ra_customer_Trx_lines_all.sales_order

used for long-term financial planning and strategy development.

58) gl source types in oracle apps?

In Oracle Apps, GL (General Ledger) source types are used to identify the source of accounting transactions.

Manual: This source type is used for transactions that are entered manually, such as journal entries or adjustments.

Interface: This source type is used for transactions that are imported from other modules in Oracle Apps, such as Accounts Receivable, Accounts Payable, or Fixed Assets.

Consolidation: This source type is used for transactions that are generated during the consolidation process. It is used to consolidate financial information from multiple business units or legal entities.

Revaluation: This source type is used for transactions that are generated during the revaluation process. It is used to adjust the value of assets and liabilities based on changes in exchange rates or other factors.

Translation: This source type is used for transactions that are generated during the translation process. It is used to convert financial information from one currency to another.

Mass Allocation: This source type is used for transactions that are generated during the mass allocation process. It is used to allocate costs or revenues to multiple accounts based on a predefined formula.

59)what is trial balance

the GL Trial Balance in Oracle Apps is a key financial report that summarizes the account balances for a specific period. It provides details on account balances, subledger balances, and allows for easy comparison of financial performance over time.

The GL Trial Balance also provides details for each account, such as the account number, name, and description

The GL Trial Balance can also include subledger balances, which provide details on specific transactions that contribute to the account balance.

The GL Trial Balance also provides drill-down capability, which allows users to view the underlying transactions that contribute to the account balance.