**Overall Goal:** The query aims to retrieve detailed information about all outgoing payments made by the company. This includes details like check numbers, supplier information, payment amounts, and even which projects the payments are related to.

**Tables Involved:**

The query pulls data from several tables in your Oracle ERP system, including:

* **AP\_CHECKS\_ALL:** Stores information about all checks issued by the company.
* **AP\_INVOICES\_ALL:** Contains details about invoices received from suppliers.
* **AP\_INVOICE\_PAYMENTS\_ALL:** Links invoices to the payments made against them.
* **AP\_SUPPLIERS:** Holds information about the company's suppliers.
* **CE\_BANK\_ACCOUNTS:** Stores details about the company's bank accounts.
* **CE\_BANK\_ACCT\_USES\_ALL:** Links bank accounts to their specific uses (e.g., payroll, expenses).
* **PA\_PROJECTS\_ALL:** Contains information about projects the company is working on.
* **FND\_FLEX\_VALUES\_TL and FND\_FLEX\_VALUES:** These tables are used for looking up descriptions for flexfields (customizable fields in Oracle applications).
* **PA\_PROJECT\_PLAYERS and PER\_ALL\_PEOPLE\_F:** These tables store information about people involved in projects and their roles.
* **RME\_PROJECT\_SECTORS:** This table seems to be a custom table (RME prefix might indicate a specific module) related to project sectors.
* **HR\_OPERATING\_UNITS:** Contains information about the company's operating units.

**Step-by-Step Explanation:**

1. **Main Query (FROM and WHERE clauses):**
   * The query starts by joining the core tables (AP\_CHECKS\_ALL, AP\_INVOICES\_ALL, AP\_INVOICE\_PAYMENTS\_ALL, AP\_SUPPLIERS, CE\_BANK\_ACCOUNTS, CE\_BANK\_ACCT\_USES\_ALL, and PA\_PROJECTS\_ALL) based on their relationships (e.g., a check has a vendor ID, which matches a supplier in the AP\_SUPPLIERS table).
   * The WHERE clause filters the data:
     + cbac.attribute1 != 'NON' likely excludes certain types of bank accounts from the results.
     + pap.PROJECT\_STATUS\_CODE in 'APPROVED' only includes payments related to approved projects.
2. **Subqueries (Nested SELECT statements):**
   * The query uses several subqueries to retrieve specific pieces of information:
     + **comp\_id:** This subquery retrieves a code combination ID from the XLA tables (related to Oracle's accounting flexfield). It seems to find the accounting code associated with the payment.
     + **cost\_center:** This subquery retrieves the description of a cost center from flexfield values based on ai.attribute7 (which likely stores a cost center code).
     + **owner:** This subquery finds the full name of the project owner from the PA\_PROJECT\_PLAYERS and PER\_ALL\_PEOPLE\_F tables.
     + **SECTOR:** This subquery retrieves the sector of a project from the custom RME\_PROJECT\_SECTORS table.
     + **VENDOR\_SITE\_CODE:** This complex CASE statement determines the vendor site code based on the supplier name. It handles special cases like 'Petty Cash', 'Staff Loan', and 'RME Deposit To Other' differently.
     + **org\_name:** This subquery gets the operating unit name from the HR\_OPERATING\_UNITS table.
3. **Selected Columns:**
   * The SELECT clause lists all the columns included in the output. It retrieves data like check number, invoice number, payment amount, supplier name, bank account details, cleared date, and various other attributes.

**Simplified Summary:**

In essence, this query retrieves comprehensive information about all cash out transactions by joining various tables related to payments, invoices, suppliers, projects, and bank accounts. It uses subqueries to fetch additional details and applies filters to narrow down the results. The query's complexity stems from the numerous tables involved and the need to handle specific cases and lookups using subqueries and CASE statements.

**To understand it better:**

* **Focus on the core tables and their relationships.** Try visualizing how the main tables connect to each other.
* **Break down the subqueries one by one.** Understand what each subquery is retrieving and how it relates to the main query.
* **Examine the CASE statement for VENDOR\_SITE\_CODE carefully.** This part seems to have custom logic for specific vendors.

If you have further questions about specific parts of the query or want to understand certain sections in more detail, feel free to ask! I'm here to help you decipher this complex SQL statement.