AI JOE

Current System Description

The PDA system is a B2B eMarketplace based in Costa Rica, currently undergoing regional expansion. Its objective is to connect buyer companies with supplier companies for the procurement of goods and services through requests for quotations (referred to as "requests") and bids (referred to as "BIT" in the database due to an error). The system manages:

- Companies (buyers and suppliers).
- Users with specific roles.
- Requests for quotations (requests).
- Items requested in each request.
- Categories to classify items.
- Supplier-category relationships (to determine whom to notify).
- Bid generation process by suppliers.
- Purchase orders derived from accepted bids.
- Product catalog for suppliers' online stores.
- Associated documentation (images, certificates, etc.).

Main Tables and Their Purpose

Table `company`

Stores company data. Key fields:

- `tax_id` (PK): Unique tax identification of the company.
- `company_name`, `company_email`, `telephone`, `contact_name`: Contact information.
- `country_id`, `city_id` (actually province/state), `country_register`: Territory associated with the company.
- `tipo_empresa` (Buyer, Supplier, Both): Defines the company's primary role in the marketplace.
- Currency, tax, advertising, mandatory documents settings, etc.

This table is essential to determine whether a company is a supplier or buyer, the territories and currencies it handles, and its general configuration.

Table `userbycompany`

Links users to companies. Key fields:

- `tax_id`: Reference to `company`.
- `type`: User role (BUYER, PROVIDER, GLOBAL ADMINISTRATOR, ADMINISTRATOR, INTERNAL BUYER). Determines the access and actions each user can perform. Currently, permissions are hardcoded, with plans for dynamic management in the future.

Table `request`

Records requests for quotations created by buyer companies. Key fields:

- `request_number` (PK, auto-increment): Unique identifier of the request.
- `tax_id`: The buyer company making the request.
- `request_date`, `request_time`: Date and time of creation.
- `request_title`: Descriptive title.
- `status`: Possible states: PENDING, OFFERING, SELECTION, POAUTHORIZING, POSENT, ACCEPT, CLOSED, CANCELED, EXPIRED.

When the request transitions from PENDING to OFFERING after authorization and categorization, relevant suppliers are notified.

Table `request_item`
Details the items requested in each request. Key fields:

- `product_id` (PK): Identifies the requested item.
- `request_number`: Links to `request`.
- `category_id`: Classifies the item into a category.
- `quantity`, `description`, `meassure_unit_id`: Defines what is being requested, in what quantity, and measurement unit.

Table `category`
Defines a hierarchy of categories (parent-child) to classify items. Key fields:

- `category_id` (PK), `code`, `parent_code`: Structure the hierarchy.

Allows the administrator to categorize requests and determine which suppliers to notify based on related child categories.

Table `provider_category`
Associates suppliers (`tax_id`) with child categories of interest (`category_id`). Key fields:

- `tax_id`
- `category_id`
- `set_c`

This determines which suppliers are notified when a request contains items from categories the supplier has marked as of interest.

Bid Tables: `global_bit` and `individual_bit`
- **`global_bit`**: Bid header. Key fields:
  - `id_bit` (PK): Bid identifier.
  - `tax_id_provider`: The supplier making the bid.
  - `request_number`: The request being responded to.

  - `global_bit_time`: Timestamp of bid creation.
  - `total`: Total amount offered.
  - `status`: ACTIVE or DESACTIVE, indicates if the bid is active or canceled by the supplier.

- **`individual_bit`**: Items offered. Key fields:
  - `id_individual_bit` (PK): Identifier of the offered item.
  - `id_global_bit`: Links to `global_bit`.
  - `id_product`: Conceptually links to `product_id` in `request_item` and `tienda_fi_productos` (see below).
  - `price`, `quantity`, `tax`, `unitary`: Details of the offered item.

 Purchase Order Tables: `global_purchasing_order` and `individual_global_purchasing_order`
When the buyer selects a bid, a purchase order is generated.

- **`global_purchasing_order`**:
  - `order_id` (PK), `order_number`, `buyer`, `provider`, `request`, `bit`: Identifies the order, links to the request and bid, and specifies the buyer and supplier involved.
  - `status`: Assigne, Procces, Rejected, Cancelled, Send Pending.

- **`individual_global_purchasing_order`**:
  - Items of the order.
  - `individual_global_purchasing_order_id` (PK), `global_purchasing_order`, `id_individual_bit`, `total`, `amount`.

 Table `tienda_fi_productos`
Supplier's product catalog for the online store. Key fields:

- `rowid` (PK): Product identifier in the catalog.
- `nombre`, `description`, `fk_categoria_tienda_1`, `fk_categoria_tienda_2`, `stock`, `unidad_medida_id`.

If marked as active, it appears in the supplier's online store. Joe will use this to find matches with requested items.

 Table `documentos_varios`
Stores paths and names of documents associated with requests or bids. Key fields:

- `rowid` (PK), `tipo`, `fk_id`, `nombre_documento`, `nombre_humano`.
- `tipo` can be "solicitud" (request) or "oferta" (bid).
- `fk_id` links to `request_number` or `id_bit`.

Here, images and certificates are stored, which Joe can use to attach to bids or analyze with OCR for more accurate matching.

New Functionality: Artificial Intelligence Agent "Joe"

Objective

"Joe" is an external AI agent that assists suppliers in creating bids. It aims to accelerate responses to requests by analyzing them with a triple approach (category, semantic analysis of name/description, recognition of texts and attached images) and offering suitable products from the supplier's catalog. Joe will be "employed" by one or more suppliers, and this relationship is established in the `ai_workforce` table with the following format:

- `tax_id` varchar(50) NOT NULL (Indicates which company is the "employer")
- `agent` set('Joe','Jane','Diana','Lucy') NOT NULL (Indicates which agent is the employee)
- `tokens` int(10) unsigned NOT NULL (Number of AI tokens consumed)
- `status` set('Activo','Inactivo') NOT NULL
- `missions` int(5) NOT NULL (Number of times a task has been requested for the employing company)

Processes with Joe

Supplier Configuration

- The supplier marks the child categories of interest in `provider_category`.
- In the `company` table, the supplier defines `country_register` and `city_id` (territories of interest).
- The supplier also configures their pricing policies, geolocation, delivery costs, etc. These policies are stored in the `company_policies` table with the structure:
  - `tax_id` varchar(50) NOT NULL
  - `geo_latitud` decimal(10,7) NOT NULL
  - `geo_longitud` decimal(10,7) NOT NULL
  - `precio_km` decimal(10,2) NOT NULL
- The supplier uploads their catalog to `tienda_fi_productos` with products, descriptions, stock, measurement units, etc.

Request Monitoring (trigger: cronjob)

- A cronjob will be implemented to periodically check `request` for new requests in PENDING status.
- Upon finding a new PENDING request, Joe authorizes and categorizes it, changing the status to OFFERING. When changed to OFFERING, relevant suppliers are notified.

Analytical Triad and Semantic Analysis

- For each item in `request_item`, Joe verifies:
  a) That the item's child category matches the categories marked in `provider_category` for the supplier (employer).
  b) That the territory is within the supplier's areas of interest.
  c) That the item's name/description, request annotations, and associated documents (in `documentos_varios`) semantically match a product in the catalog (`tienda_fi_productos`).

- Joe will apply NLP and OCR (if there are attached images) to extract text and visual information, better understanding the buyer's need. This allows finding the most suitable product.

Inventory, Measurement Units, and Pricing
- Joe will convert units if the catalog product is in a different measurement. For example, if the catalog sells in liters and the buyer requests in gallons, Joe performs the conversion. If the buyer requests 30 units and the product is sold in boxes of 20, Joe calculates the equivalent (1.5 boxes) and the resulting price according to the supplier's pricing and promotion policies.
- Joe will check the stock in `tienda_fi_productos` of his employer to ensure availability.
- If there is no inventory availability (or the requested quantity):
  - Because it is out of stock: Joe notifies the company's responsible person that the bid could not be made due to lack of inventory.
  - Because they do not sell that specific product: Joe notifies the company's responsible person that the product was requested and they cannot bid because it is not part of their inventory.
  - If the inventory quantity is insufficient, Joe will bid what is available and notify the company's responsible person that the complete bid was not made due to inventory shortage.
- Joe will consult with the supplier (via `company_policies` table or an internal API) the pricing policies, discounts, packages, or delivery costs. He can adjust the final price or create combos if he detects items commonly bought together.
- Joe will use the catalog price as a base, but in the future, he may implement dynamic pricing strategies based on sales history, seasonal demand, or other factors.

Bid Creation
- Once determined what products to offer and at what price, Joe creates the record(s) (depending on the products he can match) in `global_bit`:
  - `tax_id_provider`: of the supplier Joe works for.
  - `request_number`: the target request.
  - `global_bit_time`: Current timestamp.
  - `total`: Sum of the offered items.
  - `status`: ACTIVE.
  - Delivery time: taken from the supplier's configuration.
- In `individual_bit`, Joe inserts each item with its `id_global_bit`, `id_product` (from the catalog), `price`, `quantity`, `tax`, `unitary`, `observation`. Observations could include details about unit conversions or any clarifications.
- Attach images or certificates: Joe adds entries in `documentos_varios` with `tipo = 'oferta'` and `fk_id = id_bit` to associate relevant images or documents.

Joe's Operation Modes
- **Seller Function**: Joe bids directly. In this case:
  - Creates `global_bit` and `individual_bit`.
  - Registers in `AI_response`.
  - Inserts into `mailing` to notify the buyer immediately.

- **Sales Assistant Function**: Joe prepares the bid (simulates the creation of `global_bit` and `individual_bit` in a temporary table), but does not notify or save it definitively in `global_bit` and `individual_bit`. A user from the supplier company must review and accept the bid, making modifications if necessary. Upon acceptance:
  - The records are actually created in `global_bit` and `individual_bit`.
  - Registers in `AI_response`.
  - Inserts into `mailing` and sends the notification to the buyer.

Joe will learn from the corrections made by the human to improve future bids.

Tables `AI_response` and `mailing`
- **`AI_response`**: Fields:
  - `id` (PK)
  - `timestamp_detected`: Time when Joe detects the need.
  - `timestamp_completed`: Time when Joe completes the bid (or leaves it ready in Assistant mode).
  - `request_id`: Request number.
  - `global_bit_id`: ID of the created bid.
  - `tax_id_provider`: To know which supplier made the bid.
  - `status`: Pendiente_aprobacion, autorizado_aprobacion, autorizado, rechazado.
  - `order_id` (new field): When a purchase order is generated from this bid, the `order_id` is recorded, and the time the client took to generate the order can be calculated.
  - `id` int(32) NO PRI NULL auto_increment
  - `timestamp_detected` timestamp NO CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP
  - `timestamp_completed` timestamp YES NULL
  - `request_id` int(32) NO NULL
  - `global_bit_id` int(32) NO NULL
  - `tax_id_provider` varchar(50) NO NULL
  - `status` set('pendiente','autorizado','rechazado') NO NULL

- **`mailing`**: Fields:
  - `id` (PK)
  - `timestamp`: When the record was created.
  - `request_id`: To know which request is being notified.
  - `global_bit_id`: Bid to be communicated.
  - `id` int(32) NO PRI NULL auto_increment
  - `timestamp` timestamp NO CURRENT_TIMESTAMP
  - `request_id` int(32) NO NULL
  - `global_bit_id` int(32) NO NULL

Each new bid (in Seller mode) or each bid approved by a human (in Assistant mode) generates a record in `mailing` and triggers the sending of notifications to the buyer (email and WhatsApp).

Review of Purchase Orders (`global_purchasing_order`) and Learning
Joe reviews `global_purchasing_order` and `individual_global_purchasing_order` daily to see if his bid resulted in a sale (the buyer chose his `global_bit`).

- If the sale was made, Joe updates `AI_response` with `order_id` and a timestamp, so he knows how long it took the client to purchase.
- If Joe's bid was not selected, Joe analyzes the chosen `global_purchasing_order`:
  - Observes conditions (delivery date, warranty, price, payment term) to understand why he might have lost.
  - Notifies the supplier company that their bid was not selected, detailing the reasons, based on data such as response speed, delivery date, and an economic comparison between the selected bid and the one made by the supplier, but WITHOUT revealing amounts or the name of the selected supplier.

Confidentiality and Data Isolation
Joe can work for multiple suppliers without sharing data between them. Each instance of Joe operates with isolated data, using tokens and API-level authentication. He will never disclose sensitive data between suppliers or the name of the competition.

Integration via API
Joe is an external agent connected via a secure API. Features include:

- Authentication with JWT tokens, HTTPS encryption, roles, and permissions.
- Endpoints to notify Joe of new requests and for Joe to return bids.
- Retry mechanisms and error logging.
- Rate limiting to prevent abuse.
- Clear API documentation.

Text and Image Recognition
Joe will implement OCR to read texts in attached images and computer vision techniques to understand if an image corresponds to a certain product. This improves the semantic matching between what is requested and what the supplier has in `tienda_fi_productos`.

Learning and Strategic Recommendations
Joe will learn from experience and, based on it, make recommendations in a monthly report, detecting patterns of successful bids, demand prediction based on historical demand, inventory forecasting, pricing suggestions, and price adjustments according to seasonality.

Handling Supplier Policies
Before bidding, Joe consults the supplier's policies on pricing, packages, discounts, calculates delivery costs according to the delivery location and its own initial location, measurement units, etc. Joe will apply these policies when creating the bid.

Event-Driven Mode

If possible, implement an event-driven approach using triggers or external message queues to notify Joe in real-time when a new request is published. If not possible, maintain the cronjob. The goal is to reduce the time between request creation and notification to Joe.

Monitoring and Auditing
`AI_response` will serve as a log of Joe's actions. Dashboards can be created to monitor success rates, response times, etc. Record Joe's activities: if he corrects bids, how long the buyer takes to respond, how many bids are rejected or accepted.

Security and Privacy
- Data encryption, HTTPS communication.
- Robust authentication.
- Data isolation per supplier.
- Joe will never reveal sensitive data from another supplier or details of the winning bid from the competition.

Extensibility and Future Integrations
- Future integrations with payments, logistics, electronic invoicing.
- Action audits, detailed state change logs.
- Integration with external web services.

Dynamic Pricing and Strategies
- If bids with certain margins, measurement units, or combos are more successful, Joe will replicate that pattern.
- He will be able to predict demands, adjust prices according to seasonality or historical demand.
- Incorporate inventory forecasting, dynamic pricing, internal A/B testing.
- Adjust descriptions and bid conditions.
- A2A interaction with Jane for details and data on international deliveries, customs agents, etc.

Table Definitions (from "Tablas PDA 2025")

Table `company`
- `tax_id` varchar(50) NO PRI (Tax identifier of the company)
- `payment_id` char(2) NO
- `contact_name` varchar(50) NO
- `contact_email` varchar(50) NO
- `email_paypal` varchar(150) YES NULL
- `name_logo` varchar(50) YES NULL
- `secret_question` varchar(50) YES
- `secret_answer` varchar(50) YES
- `alert_sent` date YES NULL

- `assistance` set('Y','N') NO Y
- `company_name` varchar(40) NO (Company name)
- `company_logo` varchar(50) YES NULL
- `url` varchar(150) YES
- `address` text YES NULL
- `fax` varchar(30) YES NULL
- `country_id` varchar(11) NO (Country ID, relates to the `Country` table)
- `country_register` char(3) YES 0
- `city_id` int(11) NO 0 (ID of the region or province where the company is located)
- `telephone` varchar(40) NO (Main company phone)
- `company_email` varchar(50) NO (Main company email)
- `long_description` text YES NULL (Description of what the company does)
- `person` set('company','person') NO person
- `alerts` set('Y','N') NO N (If set to "Y", the company will receive notifications of new opportunities)
- `suggested_category_1` varchar(40) YES NULL
- `suggested_category_2` varchar(40) YES NULL
- `suggested_category_3` varchar(40) YES NULL
- `keywords` text YES NULL
- `provider_mandatory_document` varchar(50) YES NULL
- `send_provider_mandatory_document` set('Y','N') YES N
- `sponsor` set('active','inactive') NO inactive
- `start_date` date YES NULL
- `end_date` date YES NULL
- `active` set('Y','N','D') NO Y (Whether the company is active or not)
- `ranking` int(11) YES 0
- `offer_requires_authorization` set('Y','N') NO N
- `individuals_request` set('Y','N') NO Y
- `buyer_mandatory_document` varchar(50) YES NULL
- `send_buyer_mandatory_document` set('Y','N') NO N
- `delivery_address` varchar(255) YES NULL
- `payment_type` char(2) NO
- `currency` char(2) NO
- `include_taxes` set('Y','N') NO N
- `tax_porc` char(3) NO (VAT percentage applied to the company when it is a buyer)
- `unitary_prices` set('Y','N') NO N
- `minimum_offers` decimal(5,2) NO 0.00
- `deadline_days` decimal(2,0) NO 0
- `delivery_days` decimal(2,0) NO 0
- `invert_bidding` set('Y','N') NO
- `payment_mode` char(2) NO
- `default_currency` char(2) NO
- `text_by_default` text YES NULL
- `authorization_required` set('Y','N') NO N

- `po_requires_authorization` set('Y','N') NO N
- `category_set_provider` int(32) NO 2
- `category_set_buyer` int(32) NO 2
- `type_publicity` set('FL','SP','LT') NO LT
- `next_po_number` varchar(5) NO 00001
- `referer` varchar(50) YES (Tax_id of the company that invited it)
- `available_credits` int(5) NO 0
- `contact_name2` varchar(50) YES NULL (Additional contact name)
- `contact_email2` varchar(50) YES NULL (Additional contact email)
- `telephone2` varchar(40) YES NULL (Additional contact phone)
- `ext` int(3) YES NULL
- `contact_name3` varchar(50) YES NULL (Additional contact name)
- `contact_email3` varchar(50) YES NULL (Additional contact email)
- `telephone3` varchar(40) YES NULL (Additional contact phone)
- `ext3` char(3) YES NULL
- `contact_name4` varchar(50) YES NULL (Additional contact name)
- `contact_email4` varchar(50) YES NULL (Additional contact email)
- `telephone4` varchar(40) YES NULL (Additional contact phone)
- `ext4` char(3) YES NULL
- `contact_name5` varchar(50) YES NULL (Additional contact name)
- `contact_email5` varchar(50) YES NULL (Additional contact email)
- `telephone5` varchar(40) YES NULL (Additional contact phone)
- `ext5` char(3) YES NULL
- `extension_principal` varchar(10) YES NULL
- `Autorizada` set('Y','N') NO N (Whether the company is authorized to operate on the platform)
- `sms` int(1) NO 0
- `referer_link` varchar(10000) NO
- `enviar_a` int(1) NO 0
- `tipo_empresa` set('Comprador','Proveedor','Ambos') NO Ambos
- `rowid` int(10) NO MUL NULL auto_increment
- `vendedor` int(11) YES NULL
- `estado_crm` varchar(50) NO Activa
- `tienda_latitud` decimal(10,5) YES NULL
- `tienda_longitud` decimal(10,5) YES NULL
- `cabys_requerido` set('S','N') NO N
- `3cxid` varchar(200) YES NULL
- `usochat` varchar(1) YES NULL
- `subsidized` enum('Y','N') NO N
- `exporter` enum('Y','N') NO N
- `status_register` enum('COMPLETED','INCOMPLETE') YES COMPLETED

Table `userbycompany`
- `rowid` int(10) NO PRI NULL auto_increment (User ID)
- `tax_id` varchar(50) NO MUL (Tax_id of the company the user belongs to)

- `password` varchar(100) NO (User password)
- `contact_email` varchar(100) NO (User email)
- `contact_name` varchar(100) NO NULL (User name)
- `telephone` varchar(40) NO NULL (User phone)
- `personal_cellphone` varchar(40) YES NULL (User's secondary WhatsApp)
- `ext` varchar(10) NO NULL
- `amount_buyer_permit` double(5,2) YES 0.00
- `amount_provider_permit` double(5,2) YES 0.00
- `type` set('BUYER','PROVIDER','GLOBAL ADMINISTRATOR','ADMINISTRATOR','INTERNAL BUYER') NO MUL (User role)
- `status` set('ACTIVE','INACTIVE') NO (Active or inactive status)
- `notifications` enum('Y','N') NO Y (Accepts notifications or not)
- `question` varchar(255) NO
- `answer` varchar(255) NO
- `puntos` int(4) NO 0
- `remember_token` varchar(100) YES NULL
- `WhatsApp` varchar(20) YES NULL (User's primary WhatsApp)

Table `full_membership`
- `full_membership_id` int(32) NO PRI NULL auto_increment
- `tax_id` varchar(52) NO MUL (Tax_id of the company)
- `start_date` date YES 0000-00-00 (Membership start date)
- `final_date` date YES 0000-00-00 (Membership end date)
- `country` int(3) NO 0 (Country ID)
- `status` set('P','A','E','C') NO P (P: Pending, A: Active, E: Expired, C: Cancelled)

Table `request`
- `request_number` int(32) NO PRI NULL auto_increment (Request number)
- `tax_id` varchar(50) NO (Tax_id of the company)
- `contact_email_created` varchar(100) NO (Email of the user who created it)
- `currency_id` char(2) NO (Currency used)
- `request_title` varchar(80) NO (Request title)
- `deadline_date` date YES NULL (Deadline for bidding)
- `delivery_date` date NO 0000-00-00 (Delivery date)
- `request_date` date NO 0000-00-00 (Request date)
- `delivery_point` varchar(255) NO
- `payment_type` int(32) YES 0
- `currency` char(2) NO
- `garanty` varchar(1) NO N
- `extend_term` varchar(1) NO N
- `include_taxes` set('Y','N') YES Y
- `unitary_prices` set('Y','N') NO Y
- `memo` text YES NULL
- `attached_file` varchar(100) YES NULL

- `mandatory_document` set('Y','N') YES Y
- `action_ratio` int(2) NO 0
- `status`
set('OFFERING','SELECTION','POAUTHORIZING','POSENT','ACCEPT','CLOSED','CANCELED','EXPIRED','PENDING') NO OFFERING (Status. OFFERING is when it can receive bids)
- `request_time` timestamp YES CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP (Creation date and time)

Table `request_item` (Items that are part of the request)
- `product_id` int(32) NO PRI NULL auto_increment
- `product_code` char(15) NO
- `request_number` int(32) NO MUL 0 (Request number, relates to `request` table)
- `category_id` varchar(52) NO 0 (Category ID, relates to `category` table)
- `meassure_unit_id` char(2) NO (Measurement unit ID, relates to `meassure_unit` table)
- `quantity` int(32) NO 0
- `description` char(255) YES NULL

Table `meassure_unit`
- `meassure_unit_id` int(32) NO PRI NULL auto_increment
- `language` char(10) NO
- `meassure_description` char(20) NO

Table `category`
- `category_id` int(32) NO PRI NULL auto_increment
- `set_c` int(2) NO MUL 0
- `lang_id` int(32) NO 0
- `description` text NO NULL
- `request` double(48,0) YES NULL
- `offer` int(32) NO 0
- `amount` double(40,0) NO 0
- `code` varchar(15) NO (Category code)
- `parent_code` varchar(15) NO (Parent category code)
- `active_in_to` enum('Y','N') NO N

Table `provider_category`
- `set_c` int(2) NO MUL 0
- `category_id` int(32) NO MUL 0 (Relates to `category` table)
- `tax_id` varchar(52) NO MUL

Table `proveedores_invitados` (Records suppliers not in the `company` table who were invited to bid)
- `id` int(11) NO PRI NULL auto_increment
- `tax_id_invitado` varchar(52) NO NULL
- `nombre_empresa` varchar(50) NO NULL

- `correo_empresa` varchar(50) NO NULL
- `solicitud` int(32) NO NULL
- `fecha_invitada` datetime YES NULL
- `click_link` datetime YES NULL
- `oferta` datetime YES NULL
- `category_id` int(32) NO NULL
- `compradora` varchar(50) NO NULL
- `origen` set('Invitado x Compradora','Invitado Nuevo','Membresia Activa','Membresia Vencida') NO NULL
- `canal` set('Correo','WhatsApp','Llamada','Otro') NO NULL
- `agente` varchar(20) NO NULL
- `no_me_interesa` tinyint(1) NO 0

Table `proveedores_invitados_canales`
- `pk_id` int(10) NO PRI NULL auto_increment
- `fk_id_proveedores_invitados` int(11) NO NULL
- `canal` set('Correo','WhatsApp','Telefono','Web','Linkedin','Facebook','Instagram','X','Otro') NO NULL
- `valor` varchar(200) NO NULL
- `propietario` varchar(100) NO NULL

Table `PiiA`
- `pk_id` int(10) NO PRI NULL auto_increment
- `fk_request_id` int(32) YES NULL (Request ID)
- `processed` tinyint(1) YES 0 (Set to 1 when an agent starts processing it)
- `start_at` datetime YES NULL (Completed when the process starts)
- `end_at` datetime YES NULL (Completed when the process ends)
- `agent` varchar(20) YES NULL (Agent name)

Table `global_bit`
- `id_bit` int(32) NO PRI NULL auto_increment (Bid number)
- `tax_id_provider` varchar(50) NO (Tax_id of the bidding supplier)
- `tax_id_buyer` varchar(50) NO (Tax_id of the buyer)
- `request_number` int(32) NO 0 (Request number being bid on)
- `delivery_date` date NO 0000-00-00
- `date_bit` date NO 0000-00-00
- `payment_type` int(32) NO 0
- `currency_type` char(3) NO
- `observation` text NO NULL
- `tax_porc` char(3) NO
- `tax` char(3) NO
- `unitary_price` set('Y','N') NO
- `pay_deadline` int(10) YES NULL
- `garanty_off` int(10) YES NULL

- `document` set('Y','N') NO Y
- `offer_file` varchar(150) NO
- `total` double(10,2) NO 0.00
- `status` set('ACTIVE','DESACTIVE') NO ACTIVE
- `read_offer` set('Y','N') NO N
- `global_bit_time` timestamp YES NULL
- `gravado` double(10,2) NO NULL
- `exento` double(10,2) NO NULL
- `monto_impuesto` double(10,2) NO NULL

Table `individual_bit` (Items of the `global_bit` bid)
- `id_individual_bit` int(32) NO PRI NULL auto_increment
- `id_global_bit` int(32) NO MUL 0 (Bid number)
- `id_product` int(32) NO 0
- `category` int(32) NO 0
- `price` double(10,2) NO 0.00
- `quantity` int(15) NO 0
- `tax` set('Y','N') NO
- `unitary` set('Y','N') NO
- `observation` text NO NULL
- `check_exento` char(1) NO NULL
- `cabys` varchar(13) NO NULL
- `tax_porc` int(3) NO 0

Table `global_purchasing_order` (Purchase orders)
- `order_id` int(32) NO PRI NULL auto_increment
- `order_number` varchar(5) NO 00000 (Purchase order number)
- `order_date` date NO 0000-00-00
- `buyer` varchar(50) NO 0 (Tax_id of the buyer)
- `provider` varchar(50) NO 0 (Tax_id of the supplier assigned the purchase order)
- `request` int(32) NO 0 (Request number)
- `bit` int(32) NO 0 (Bid number / `global_bit`)
- `status` set('Assigne','Procces','Rejected','Cancelled','Send Pending') NO Procces
- `global_purchasing_order_time` timestamp NO CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP

Table `individual_global_purchasing_order`
- `individual_global_purchasing_order_id` int(32) NO PRI NULL auto_increment
- `global_purchasing_order` int(32) NO MUL 0 (Purchase order number)
- `id_individual_bit` int(32) NO 0
- `total` double(12,2) NO 0.00
- `amount` int(32) NO 0

Table `login`

- `id` int(11) NO PRI NULL auto_increment
- `user_id` int(11) NO NULL (User ID from `userbycompany` table)
- `login_time` timestamp NO CURRENT_TIMESTAMP