

### Overview

The Band Gear Exchange Forum is a web-based service that allows users, referred to as “providers”, to create and accept offers for exchanging band equipment. This document outlines the functionalities, user types, and operations of the service.

### User Registration

Providers can register and list their exchange offers in various categories, such as:

- Instruments
- Amplifiers
- Microphones
- Studio Equipment
- Etc.
- 

### Offer Details

Providers input the details of their exchange offers as follows:

- Define the start and end dates of the exchange offer.
- Set compensation and delivery preferences.
- Maintain the status of their exchange offers and adjust the content as needed.
- Remove offers once the exchange is completed.

### System Functions

The system maintains, searches, suggests, and sorts exchange offers for band equipment. It includes:

- Creation and maintenance of band equipment lists.
- Maintenance and printing of open offers lists.
- Status tracking of offers.

### Listing and Status Management

- Lists offers by provider, status, category, date, and content.
- Changes the status of offers that lead to exchanges to “completed” but does not delete them.

### Database Entities

#### Provider Information

- **User ID:** Unique identifier for the provider.
- **Name:** Provider’s name.

- **Nickname:** Provider's nickname.
- **Password:** Provider's password.
- **Email:** Provider's email address.
- **Phone Number:** Visible only to registered users.

## Offer-Related Customer Information

### Offer Group Table

- **Offer Group Code:** Unique code for the offer group.
- **Offer Group Name:** Name of the offer group.

### Offer Content Table

- **Offer ID:** Unique identifier for the offer.
- **Offer Status:** Status of the offer.
- **Number of Equipment Rows:** Number of equipment rows in the offer.
- **Equipment Row Details:**
  - **Quantity Offered:** Quantity of the equipment offered.
  - **Offer Price:** Price of the equipment offered.
  - **Vintage Product Life Story** (optional): History and background of the vintage product.
  - **Original Price:** Original price of the product.
  - **Comments and Additional Information:** Any additional details about the offer.
  -

### Exchange Offer Table ("Purchase Offer")

- **Offer ID:** Unique identifier for the offer.
- **Offer Status:** Status of the offer.
- **Provider ID:** Unique identifier for the provider.
- **Provider Contact Information:** Contact details of the provider.
- **Provider Preferences and References:** Preferences and references of the provider.
- **Start and End Dates:** Start and end dates of the offer.
- **Number of Product Rows:** Number of product rows in the offer.
- **Offered Equipment IDs:** IDs of the offered equipment.
- **Delivery Method:** Method of delivery for the equipment.
- **Exchange Preferences:** Desired products for exchange.

- **Additional Offer Information:** Any other relevant details about the offer.

### Equipment Group Table

- **Equipment Group Code:** Unique code for the equipment group.
- **Equipment Group Name:** Name of the equipment group.

### Equipment Table

- **Equipment Code:** Unique code for the equipment.
- **Equipment Name and Type Details:** Name and type details of the equipment, such as model year.
- **Detailed Description:** Detailed description of the equipment, including condition, color, sound, etc.

### Historical Statistics and Views

- **Yearly and Monthly Exchange Statistics by Product Group:** Statistics on exchanges categorized by product group for each year and month.
  - **Provider Statistics and User List:** Statistics on providers and a list of users.
- 

## Database and Conceptual Model Documentation

### Introduction

The following sections provide detailed descriptions of each table and their relations.

### Database Schema

#### 1. Bidder Table

The Bidder table stores information about users who participate in the bidding system.

- **id:** A unique identifier for each bidder (Primary Key).
- **bidderName:** The name of the bidder (must be unique).
- **password:** The password for the bidder's account (optional).
- **admin:** A boolean flag indicating if the bidder has administrative privileges (default is FALSE).

```
CREATE TABLE Bidder
(
    id SERIAL NOT NULL PRIMARY KEY,
    bidderName VARCHAR(30) NOT NULL UNIQUE,
    password VARCHAR(20) DEFAULT NULL,
    admin BOOLEAN DEFAULT FALSE
);
```

#### 2. BidGenre Table

The BidGenre table categorizes the types of bids.

- **id:** A unique identifier for each bid genre (Primary Key).
- **bidGenre:** An integer representing the genre.

- **bidder**: A foreign key referencing the Bidder table.
- **bidGenreHeader**: A unique header for the bid genre.

```
CREATE TABLE BidGenre
(
    id SERIAL NOT NULL PRIMARY KEY,
    bidGenre INT,
    bidder INT,
    bidGenreHeader VARCHAR(100) NOT NULL UNIQUE,
    FOREIGN KEY(bidder) REFERENCES Bidder(id)
);
```

### 3. BidHeader Table

The BidHeader table contains information about individual bids.

- **id**: A unique identifier for each bid header (Primary Key).
- **initialBidder**: A foreign key referencing the Bidder table.
- **bidGnre**: A foreign key referencing the BidGenre table.
- **bidHder**: A unique header for the bid.
- **numberOfBids**: The number of bids made.
- **startedTime**: The timestamp when the bid started.

```
CREATE TABLE BidHeader
(
    id SERIAL NOT NULL PRIMARY KEY,
    initialBidder INT,
    bidGnre INT,
    bidHder VARCHAR(100) NOT NULL UNIQUE,
    numberOfBids INT NOT NULL,
    startedTime TIMESTAMP NOT NULL,
    FOREIGN KEY(initialBidder) REFERENCES Bidder(id),
    FOREIGN KEY(bidGnre) REFERENCES BidGenre(id)
);
```

### 4. BidContent Table

The BidContent table stores the content of individual bids.

- **id**: A unique identifier for each bid content (Primary Key).
- **bidderID**: A foreign key referencing the Bidder table.
- **bidHead**: A foreign key referencing the BidHeader table.
- **bidderName**: The name of the bidder.
- **bidContent**: The content of the bid.
- **editedTime**: The timestamp when the bid was last edited.

```
CREATE TABLE BidContent
(
    id SERIAL NOT NULL PRIMARY KEY,
```

```

    bidderID INT,
    bidHead INT,
    bidderName VARCHAR(30) NOT NULL,
    bidContent VARCHAR(500) NOT NULL,
    editedTime TIMESTAMP NOT NULL,
    FOREIGN KEY(bidHead) REFERENCES BidHeader(id)
);

```

## 5. BidExchange Table

The BidExchange table manages the exchanges between bidders.

- **id**: A unique identifier for each bid exchange (Primary Key).
- **salesBidder**: A foreign key referencing the Bidder table (initial bidder).
- **purchaseBidder**: A foreign key referencing the Bidder table (purchasing bidder).
- **equipmentOffer**: A description of the equipment offered.
- **equipmentRequested**: A description of the equipment requested.
- **agreedDeal**: A boolean flag indicating if the deal is mutually agreed upon (default is FALSE).
- **dealTimestamp**: The timestamp when the deal was made.

```

CREATE TABLE BidExchange
(
    id SERIAL NOT NULL PRIMARY KEY,
    salesBidder INT NOT NULL,
    purchaseBidder INT NOT NULL,
    equipmentOffer VARCHAR(100) NOT NULL,
    equipmentRequested VARCHAR(100) NOT NULL,
    agreedDeal BOOLEAN DEFAULT FALSE,
    dealTimestamp TIMESTAMP NOT NULL,
    FOREIGN KEY(salesBidder) REFERENCES Bidder(id),
    FOREIGN KEY(purchaseBidder) REFERENCES Bidder(id)
);

```

The relationships between the tables are as follows:

- **Bidder**: Represents users in the system. Each bidder can create multiple bid genres and bid headers.
- **BidGenre**: Categorizes bids and is linked to a specific bidder.
- **BidHeader**: Contains the main information about a bid, including the initial bidder and the genre of the bid.
- **BidContent**: Stores the detailed content of each bid and is linked to both the bidder and the bid header.
- **BidExchange**: Manages the exchanges between bidders, including the details of the equipment offered and requested.

```

CREATE TABLE Bidder
(
    id SERIAL NOT NULL PRIMARY KEY,

```

```
bidderName VARCHAR(30) NOT NULL UNIQUE,  
password VARCHAR(20) DEFAULT NULL,  
admin BOOLEAN DEFAULT FALSE  
);
```

```
CREATE TABLE BidGenre  
(  
    id SERIAL NOT NULL PRIMARY KEY,  
    bidGenre INT,  
    bidder INT,  
    bidGenreHeader VARCHAR(100) NOT NULL UNIQUE,  
  
    FOREIGN KEY(bidder) REFERENCES Bidder(id)  
);
```

```
CREATE TABLE BidHeader  
(  
    id SERIAL NOT NULL PRIMARY KEY,  
    initialBidder INT,  
    bidGnre INT,  
    bidHder VARCHAR(100) NOT NULL UNIQUE,  
    numberOfBids INT NOT NULL,  
    startedTime TIMESTAMP NOT NULL,  
  
    FOREIGN KEY(initialBidder) REFERENCES Bidder(id),  
    FOREIGN KEY(bidGnre) REFERENCES BidGenre(id)  
);
```

```
CREATE TABLE BidContent  
(  
    id SERIAL NOT NULL PRIMARY KEY,  
    bidderID INT,  
    bidHead INT,  
    bidderName VARCHAR(30) NOT NULL,  
    bidContent VARCHAR(500) NOT NULL,  
    editedTime TIMESTAMP NOT NULL,  
  
    FOREIGN KEY(bidHead) REFERENCES BidHeader(id)  
);
```

```
CREATE TABLE BidExchange  
(  
    id SERIAL NOT NULL PRIMARY KEY,  
    salesBidder INT NOT NULL, -- ID of the initial bidder (sales-bidder)  
    purchaseBidder INT NOT NULL, -- ID of the bidder making the purchase offer
```

```

equipmentOffer VARCHAR(100) NOT NULL, -- Description of the equipment offered
equipmentRequested VARCHAR(100) NOT NULL, -- Description of the equipment requested
agreedDeal BOOLEAN DEFAULT FALSE, -- Indicates if the deal is mutually agreed upon
dealTimestamp TIMESTAMP NOT NULL, -- Timestamp when the deal was made

FOREIGN KEY(salesBidder) REFERENCES Bidder(id),
FOREIGN KEY(purchaseBidder) REFERENCES Bidder(id)
);

--Explanation for table BidExchange:
--salesBidder and purchaseBidder columns store the IDs of the bidders involved in the
exchange.
--equipmentOffer describes the equipment offered by the sales bidder.
--equipmentRequested describes the equipment requested by the purchaser.
--agreedDeal indicates whether both parties have agreed on the deal (initially set to
FALSE).
--dealTimestamp records the time when the deal was made.
--Remember to adjust the column definitions according to your specific requirements.
--This table will allow bidders to exchange equipment based on their bids.

CREATE VIEW BidGenreList AS
SELECT bg.id, bg.bidGenreHeader, COUNT(bc.id) AS count, MAX(bc.editedTime) AS latest
FROM BidGenre bg
LEFT JOIN BidHeader bhr ON (bg.id = bhr.bidGnre)
LEFT JOIN BidContent bc ON (bhr.id = bc.bidHead)
GROUP BY bg.id, bg.bidGenreHeader;

CREATE VIEW BidList AS
SELECT bh.id, bh.initialBidder, bh.bidGnre, bh.bidHder,
COUNT(*) AS count, MIN(bc.editedTime) AS edited,
MAX(bc.editedTime) AS latest
FROM BidHeader bh
LEFT JOIN BidContent bc ON (bh.id = bc.bidHead)
GROUP BY bh.id, bh.bidHder, bh.initialBidder, bh.bidGnre
ORDER BY MAX(bc.editedTime) DESC;

```