BROKER

Purpose: Allows storage and retrieval of Broker information

How BROKER is used: Users will SELECT, INSERT, UPDATE and DELETE broker information.

When BROKER is used: Data is inserted when a new stock broker added.

BROKER PRIMARY KEY (broker_id)			
broker_id	NUMBER(6)	NOT NULL	ID for the broker
first_name	VARCHAR2(25)	NOT NULL	First Name of the Broker
last_name	VARCHAR2 (25)	NOT NULL	Last Name of the Broker

BROKER_STOCK_EX

Purpose: Records associations between brokers and stock exchanges.

How BROKER_STOCK_EX is used: Users will SELECT, INSERT, UPDATE, and DELETE broker information.

When BROKER_STOCK_EX is used: BROKER_STOCK_EX stores a link between a broker and a stock_exchange if that broker may trade on the exchange.

BROKER PRIMARY KEY (stock_ex_id, broker_id)			
stock_ex_id	NUMBER(6)	NOT NULL	ID for the Stock Exchange
broker_id	NUMBER(6)	NOT NULL	ID for the broker

COMPANY

Purpose: Storage of both institutional shareholders as well as companies whose stocks are traded.
How COMPANY is used: The company table is a subtype of SHAREHOLDER. Users will SELECT, INSERT, UPDATE and DELETE Company
When COMPANY is used: Data is inserted for either companies whose stock is traded, or for institutional shareholders. If a company is publicly traded then the stock_id, starting_price, and currency_id cannot be NULL.

COMPANY PRIMARY KEY (company_id)			
company_id	NUMBER(6,0)	NOT NULL	ID for the company
name	VARCHAR2 (20)	NOT NULL	Name for the company
place_id	NUMBER (6,0)	NOT NULL	Location of the company
stock_id	NUMBER (6,0)	NULL	ID for the stock of the company. NULL if the company is not traded.
starting_price	NUMBER(10,4)	NULL	Initial price of stock when it is offered as part of the IPO.
currency_id	NUMBER(6,0)	NULL	ID of the currency used for the starting price, and the currency used for the price_total of any trades on the primary market.

CONVERSION

Purpose: Provides information to convert between currencies.
How CONVERSION is used: CONVERSION is actually a VIEW. Users will SELECT from the view but must perform INSERTs, UPDATEs and DELETEs on the CONVERTION_RATE table.
When CONVERSION is used: CONVERSION could be queried at any time.

CONVERSION PRIMARY KEY (from_currency_id, to_currency_id)			
from_currency_id	NUMBER(6)	NOT NULL	ID of the currency converted FROM
to_currency_id	NUMBER(6)	NOT NULL	ID of the currency converted TO
exchange_rate	NUMBER(10,4)	NOT NULL	Exchange rate

CONVERSION_RATE

Purpose: Stores currency exchange rates.
How CONVERSION_RATE is used: The CONVERTION_RATE table stores exchange rates but stores the exchange rate for only one direction. For example, the table stores the exchange rate from US Dollars to British Pounds but does not store the exchange rate from British Pounds to US Dollars.
When CONVERSION_RATE is used: The CONVERSION_RATE table is queries by the CONVERSION view. Updates to the data are made directly to the CONVERSION_RATE table and could be made at any time.

CONVERSION_RATE PRIMARY KEY (from_currency_id, to_currency_id)			
from_currency_id	NUMBER(6)	NOT NULL	ID of the currency converted FROM
to_currency_id	NUMBER(6)	NOT NULL	ID of the currency converted TO
exchange_rate	NUMBER(10,4)	NOT NULL	Exchange rate

CURRENCY

Purpose: Allows the storage and retrieval of currency information.
How CURRENCY is used: Users will SELECT, INSERT, UPDATE and DELETE currency.
When CURRENCY is used: Data is inserted when a new currency must be added into the system. The CURRENCY table is referenced by the CONVERTION_RATE table, the STOCK_EXCHANGE table and the COMPANY table.

CURRENCY PRIMARY KEY (currency_id)			
currency_id	NUMBER(6)	NOT NULL	ID for the Currency
name	VARCHAR2(5)	NOT NULL	Name of the currency
symbol	VARCHAR2 (50)	NOT NULL	Symbol that represents the currency

CURRENT_SHAREHOLDER_SHARES

Purpose: Allows retrieval of the current number of shares held by any specific shareholder.
How CURRENT_SHAREHOLDER_SHARES is used: CURRENT_SHAREHOLDER_SHARES is actually a view. Users will SELECT from the view but, must perform INSERTs, UPDATEs and DELETEs on the TRADE table.
When CURRENCT_SHAREHOLDER_SHARES is used: CURRENT_SHAREHOLDER_SHARES could be queried at any time.

CURRENT_SHAREHOLER_SHARES PRIMARY KEY (none)			
shareholder_id	NUMBER	NULL	ID for the Shareholder
type	VARCHAR2(25)	NOT NULL	The type of shareholder. Could be "DIRECT_HOLDER", or "COMPANY"
stock_id	NUMBER	NULL	ID for stock being held
shares	NUMBER	NULL	Number of shares held by the shareholder

CURRENT_STOCK_STATS

Purpose: Allows retrieval of statistics about all current stocks.
How CURRENT_STOCK_STATS is used: CURRENT_STOCK_STATS is actually a view. Users will SELECT from this view, but must perform INSERTs, UPDATES, and DELETES on the SHARES_AUTHORIZED and TRADE tables.
When CURRENT_STOCK_STATS is used: CURRENT_STOCK_STATS could be queried at any time.

CURRENT_STOCK_STATS PRIMARY KEY (none)			
stock_id	NUMBER(6)	NULL	ID for the stock
current_authorized	NUMBER(12,4)	NOT NULL	Current authorized shares
total_outstanding	NUMBER	NULL	Total shares outstanding

DIRECT_HOLDER

Purpose: Allows storage and retrieval of direct holder information.
How DIRECT_HOLDER is used: The DIRECT_HOLDER table is a subtype of SHAREHOLDER. User will SELECT, INSERT, UPDATE and DELETE direct holder.
When DIRECT_HOLDER is used: Data is inserted whenever a shareholder is added to the system.

DIRECT_HOLDER PRIMARY KEY (direct_holder_id)			
direct_holder_id	NUMBER(6,0)	NOT NULL	ID for the direct holder
first_name	VARCHAR2(25)	NOT NULL	First Name for the direct holder
last_name	VARCHAR2(25)	NOT NULL	Last Name for the direct Holder

PLACE

Purpose: Allows storage and retrieval of place information.
How PLACE is used: Users will SELECT, INSERT, UPDATE and DELETE place table.
When PLACE is used: PLACE could be queried at any time.

PLACE PRIMARY KEY (place_id)			
place_id	NUMBER(6)	NOT NULL	ID for Place
city	VARCHAR2(50)	NOT NULL	Name of city
country	VARCHAR2(50)	NOT NULL	Name of country

SHAREHOLDER

Purpose: The SHAREHOLDER table is the supertype for the COMPANY and DIRECT_HOLDER subtypes.
How SHAREHOLDER is used: Users will SELECT, INSERT, UPDATE and DELETE shareholder
When SHAREHOLDER is used: Data is inserted whenever a new shareholder is identified. If the TYPE is “COMPANY” then there must a corresponding entry in the COMPANY table. If the TYPE is “DIRECT_HOLDER” then there must be a corresponding entry in the DIRECT_HOLDER table.

SHAREHOLDER PRIMARY KEY (shareholder_id)			
shareholder_id	NUMBER(6,0)	NOT NULL	ID for the shareholder
type	VARCHAR2(25)	NOT NULL	Identify if shareholder is company or direct holder

SHARES_AUTHORIZED

Purpose: Allows storage and retrieval of authorized shares
How SHARES_AUTHORIZED is used: User will SELECT, INSERT, UPDATE and DELETE shares authorized
When SHARES_AUTHORIZED is used: When a company issues stock, it must declare the total number of shares authorized.
If a company changes the number of shares authorized then any existing SHARES_AUTHORIZED record must be expired, and a new record must be entered. The TIME_START indicates when the record comes into effect, and the TIME_END indicate when the record is no longer in effect. A TIME_END of NULL indicates that the record has not expired.
SHARES_AUTHORIZED records for a single STOCK_ID must not overlap in time. Only one record for each STOCK_ID can have a null TIME_END.

SHARES_AUTHORIZED PRIMARY KEY (stock_id, time_start)			
stock_id	NUMBER(6,0)	NOT NULL	ID for the stock
time_start	DATE	NOT NULL	Date of when stocks were first available
time_end	DATE	NULL	Expiry date for the record. When a company changes its shares authorized, the existing record is expired and a new record is entered.
authorized	NUMBER (12,4)	NOT NULL	Number of shares authorized for trading

STOCK_EXCHANGE

Purpose: Allows storage and retrieval of stock exchanges.
How STOCK_EXCHANGE is used: Users will SELECT, INSERT, UPDATE and DELETE stock_exchange table.
When STOCK_EXCHANGE is used: STOCK_EXCHANGE could be queried at any time.

STOCK_EXCHANGE **PRIMARY KEY (stock_ex_id)**			
stock_ex_id	NUMBER(6)	NOT NULL	ID for the Stock Exchange
name	VARCHAR2(50)	NOT NULL	Name for the Stock Exchange
symbol	VARCHAR2(10)	NOT NULL	Symbol for the Stock Exchange
currency_id	NUMBER(6)	NOT NULL	ID for the currency
place_id	NUMBER(6)	NOT NULL	ID for the location of Stock Exchange

STOCK_LISTING

Purpose: Allows storage and retrieval of Stock/ Stock Exchange relationship; as well as its given stock symbol.
How STOCK_LISTING is used: It stores the stock symbol as well as the relationship between stock and stock exchange table.
When STOCK_LISTING is used: Data is inserted when a stock is listed on any given stock exchange. The stock is also assigned a unique symbol that will reference the specific stock.

STOCK_LISTING PRIMARY KEY (stock_ex_id, stock_id)			
stock_ex_id	NUMBER(6)	NOT NULL	ID for the Stock Exchange
stock_id	NUMBER(6)	NOT NULL	ID for the Stock
stock_symbol	VARCHAR(20)	NOT NULL	Symbol that is used by the stock on the particular stock exchange

STOCK_PRICE

Purpose: Allows storage and retrieval of stock price
How STOCK_PRICE is used: Users will SELECT, INSERT, UPDATE and DELETE stock price.
When STOCK_PRICE is used: The market price must be recorded in the STOCK_PRICE table before a stock can be traded on a stock exchange. When the market price changes, the actual record must be expired and a new record must be inserted to record the new market price.

STOCK_PRICE PRIMARY KEY (stock_ex_id, stock_id, time_start)			
stock_ex_id	NUMBER(6)	NOT NULL	ID for the Stock Exchange
stock_id	NUMBER(6)	NOT NULL	ID for the Stock
time_start	DATE	NOT NULL	Date and time when the price became the active price
time_end	DATE	NULL	Date and time when the price stopped being active. NULL indicates that the price is the currently active price. Only one row can have a NULL time_end for any (stock_ex_id, stock_id) pair.
price	NUMBER(10,4)	NOT NULL	Price of an individual share of stock.

TRADE

Purpose: Allows storage and retrieval of trade information for trades on both the primary market and the secondary markets. Trade records can also indicate adjustments that are made during stock splits and merges.
How TRADE is used: Users will SELECT, INSERT, UPDATE and DELETE trade information.

TRADE PRIMARY KEY (trade_id)			
trade_id	NUMBER(9)	NOT NULL	ID for the Trade
stock_id	NUMBER(6)	NOT NULL	ID for the Stock
transaction_time	DATE	NOT NULL	Date and time when the trade occurred
shares	NUMBER(12,4)	NOT NULL	Number of Shares traded
stock_ex_id	NUMBER(6)	NULL	Id of the Stock Exchange. STOCK_EX_ID is null for trades on the primary market.
price_total	NUMBER(20,2)	NULL	Total Price. When shares are acquired or removed due to a stock split or merge the price_total is NULL.
buyer_id	NUMBER(6)	NOT NULL	Shareholder_ID of the buyer
seller_id	NUMBER(6)	NOT NULL	Shareholder-ID of the seller
buy_broker_id	NUMBER(6)	NULL	ID of the broker buying the stock. For trades on the primary market and for adjustments due to stock splits and merges, the buy_broker_id is NULL.
sell_broker_id	NUMBER(6)	NULL	ID of the broker selling the stock. For trades on the primary market and for adjustments due to stock splits and merges, the sell_broker_id is NULL