دانشگاه صنعتی شریف

پروژه‌ی صرافی ارز دیجیتال: پایگاه داده - فاز اول

استاد: دکتر مهدی آخی

شمارهٔ تیم: ۱۴

مانی ابراهیمی	محمدامین حیدری	محمد جعفری‌پور
۴۰۱۱۷۰۴۹۱	۴۰۱۱۷۰۵۵۳	۴۰۱۱۰۵۷۹۷

بهار ۱۴۰۳

فهرست مطالب

فصل ۱

کلیت فاز اول پروژه

۱.۱ شرح

در این فاز تلاش شده تا یک پایگاه داده‌ی مرتبط با یک صرافی ارز دیجیتال طراحی شود. این پایگاه داده شامل موجودیت‌هایی مانند کاربر و کیف پول و تراکنش است. همچنین برای هر موجودیت روابطی با موجودیت‌های دیگر نیز تعریف شده است. در ادامه به توضیح هر یک از موجودیت‌ها و روابط آن‌ها با موجودیت‌های دیگر پرداخته‌ایم. همچنین در انتها پاسخ به ۱۰ پرسش جبر رابطه‌ای داده شده نیز آمده است. مخزن یا همان repository **این پروژه در اینجا[1] قابل مشاهده است.**

۱.۲ تقسیم وظایف

تیم این پروژه متشکل از سه نفر بود که برای سادگی در سند تقسیم وظایف، برای آن‌ها از اسم کوتاه استفاده کردیم:

نام کوتاه	نام کامل	شماره دانشجویی
Mani	مانی ابراهیمی	۴۰۱۱۷۰۴۹۱
Mamadamin	محمدامین حیدری	۴۰۱۱۷۰۵۵۳
Mamal	محمد جعفری‌پور	۴۰۱۱۰۵۷۹۷

جدول ۱.۱: جدول اعضای تیم در جدول تقسیم وظایف

جدول تقسیم وظایف نیز از اینجا[2] قابل مشاهده است.

[1] **در صورتی که لینک برای شما کار نمی‌کند، از آدرس** https://github.com/maniebra/dbms-exchange-project **استفاده نمایید.**

[2] **در صورتی که این لینک برای شما کار نمی‌کند، میتوانید از آدرس**
https://docs.google.com/spreadsheets/d/1x1Guh4HTWLyG9GTomZEtesp5cjIGez9m9Day3bS_kgM/edit?usp=sharing
استفاده نمایید.

فصل ۲

دیاگرام های ER

۲.۱ شرح

در این بخش تلاش بر این بود که کلیت پایگاه داده‌ی مورد نظر را با استفاده از دیاگرام‌های ER نمایش دهیم. ابتدا دیاگرام ER اصلی را نمایش داده‌ایم و سپس به تفکیک بخش‌های مختلف آن پرداخته‌ایم.

۲.۲ توضیح هر موجودیت

در ادامه، برای هر موجودیت حاضر در این دیاگرام توضیحی آمده:

۲.۲.۱ User

موجودیت کاربر یا همان user، که دارای صفات گفته شده از جمله نام و نام خانوادگی و شناسه ملی و شماره تماس و ایمیل و رمز عبور و سایر موارد است. این موجودیت برای کاربران اصلی‌ترین موجودیت بوده چرا که اطلاعات خود هر کاربر را در این موجودیت ذخیره می‌کنیم. همچنین به یک موجودیت کیف پول متصل است که باعث می‌شود هر کاربر یک کیف پول داشته باشد.

۲.۲.۲ Wallet

موجودیت کیف پول یا همان wallet، که دارای صفات گفته شده از جمله موجودیت کاربر و موجودی و ارز و سایر موارد است. این موجودیت برای ذخیره‌ی اطلاعات مربوط به کیف پول هر ارز از هر کاربر استفاده می‌شود. همچنین به یک موجودیت تراکنش متصل است که باعث می‌شود هر کیف پول دارای تراکنش باشد.

۲.۲.۳ Transactions

موجودیت تراکنش یا همان transactions، که دارای صفات گفته شده از جمله موجودیت کیف پول و نوع تراکنش و مبلغ و تاریخ و سایر موارد است. این موجودیت برای ذخیره‌ی اطلاعات مربوط به تراکنش‌های هر کیف پول استفاده می‌شود. در هر تراکنش مقداری ارز از کیف پول یک کاربر خارج شده و به کیف پول کاربری دیگر می‌رود. در نظر داشته باشید که هر تبادل، دو تراکنش است.

همچنین صفت fee در تراکنش با داشتن Market_id و بدست آوردن ارز پایه‌ی آن مارکت و قیمت لحظه‌ای آن ارز پایه به ریال محاسبه می‌شود.

۲.۲.۴ Orders

موجودیت سفارش ها یا همان Orders، که دارای صفات گفته شده از جمله تاریخ و وضعیت و نوع ارز و حجم و قیمت و سایر موارد است. این موجودیت برای ذخیره اطلاعات مربوط به سفارشات کاربران می باشد و دارای دو نوع خرید و فروش می باشد. همچنین به یک موجودیت تبادل متصل است در اصل ترکیب دو سفارش خرید و فروش می باشد.

۲.۲.۵ Trades

موجودیت تبادل ها یا همان Trades، که دارای صفات گفته شده از جمله تاریخ و حجم و مقدار و سایر موارد است. این موجودیت برای ذخیره اطلاعات مربوط به تبادل ها می باشد که تبادل ها میتواند بین یک کاربر و ادمین سایت و یا دو کاربر باشد که به ترتیب دو موجودیت OTC و P2Pرو تشکیل داده اند. همچنین این موجودیت دارای یک شناسه برای هر تبادل می باشد. صفت `min_fill_remainder` به این صورت عمل می‌کند که حجم باقی‌مانده‌ی کمینه‌ی دو سفارش خرید و فروش را ذخیره می‌کند.

OTC

شامل ID ادمین و مشتری می‌باشد که بوسیله‌ی شناسه‌ی Market به بازار مربوطه متصل شده است.

P2P

شامل دو ID و OrderID خریدار و فروشنده یا همان maker و taker می‌باشند که به وسیله‌ی شناسه‌ی صرافی یا همان Broker_ID به صرافی مربوطه متصل شده‌اند.

۲.۲.۶ OrderBooks

موجودیت لیست سفارشات یا همان OrderBooks، که دارای صفات گفته شده از جمله شناسه و شناسه ی بازار و و سایر موارد است. این موجودیت برای ذخیره اطلاعات مربوط به لیست های سفارشات هر فروشگاه میباشد، همچنین به یک موجودیت لیست که زیرمجموعه‌ی OrderBooks است متصل شده که شامل دو نوع لیست خرید و فروش می باشد و به موجودیت سفارشات که خود دو نوع خرید و فروش دارد نیز متصل است که در نهایت این دو نوع خرید و فروش با هم سفارشات را بتواند بسازد.

۲.۲.۷ Markets

موجودیت فروشگاه ها یا همان Markets، که دارای صفات گفته شده از جمله کارمزد و قیمت لحظه ای بازار و نوع ارز پایه و سایر موارد است. این موجودیت برای ذخیره اطلاعات مربوط به فروشگاه های خرید و فروش ارز دیجیتال برای کاربران می باشد. همچنین به یک موجودیت لیست سفارشات متصل است که شامل دو لیست خرید و فروش هر فروشگاه می باشد.

۲.۲.۸ Brokers

موجودیت صرافی ها یا همان Brokers، که دارای صفات گفته شده از جمله شناسه و سایر موارد است. این موجودیت برای ذخیره اطلاعت مربوط به صرافی های ارز دیجیتال می باشد. همچنین به موجودیت فروشگاه ها متصل می باشد که برای هر ارز پایه در صرافی یک فروشگاه وجود دارد و به یک یا چند admin متصل است که در ان ادمین های هر صرافی مشخص می شوند.

۲.۲.۹ CryptoCurrency

موجودیت کریپتو ها یاCryptoCurrency ارز هایی اند که در سایت وجود دارند و توسط افراد مبادله میشوند. این ارز ها ممکن است قیمت ثابت Stable coinباشند و یا قیمت انها هرلحظه عوض شود nonstable Currency.

۲.۲.۱۰ Network

هر ارز شامل چندین شبکه ی مجزا از هم است که تراکنشهای آنها روی بستر متفاوتی انحام میشود.این شبکه ها دارای کارمزد و زمان متفاوتی اند.

۲.۲.۱۱ Online Payments

تاریخچه‌ی تمامی واریزی‌های هر کاربر، مقدار آن و زمان انجام‌شده است.

۲.۲.۱۲ Wallet History

تاریخچه‌ای از تغییرات میزان هر کیف پول است و هر تراکنشی برای دو کیف پول یک Wallet History جدید می‌سازد.

۲.۲.۱۳ Crypto Histories

تاریخچه‌ی تغییرات قیمت یک رمزارز است که زمان آن تغییر و مقدار و قیمت آن در آن زمان (قیمت همان قیمت لحظه‌ای مارکت است) نشان می‌دهد. با انحام هر تراکنش یک CryptoHistory جدید ایجاد می‌شود چرا که قیمت لحظه‌ای ارز تغییر می‌کند.

فصل ۳

سوالات جبر رابطه‌ای

۳.۱ شرح

در این بخش پاسخ به ۱۰ سوال جبر رابطه‌ای[۱] آمده است.

۳.۲ پاسخ به سوالات

۳.۲.۱ سوال ۱

$$\Pi_{market\_id, fee}(Transactions \bowtie_{Transactions.market\_id=Market.market\_id \wedge Transactions.date=date} (\ _{market\_id}\mathcal{F}_{\max(date)}(Market \bowtie_{Market.market\_id=Transactions.market\_id} Transactions)))$$

۳.۲.۲ سوال ۲

$$_{owner\_id}\mathcal{F}_{\texttt{Sum}(total\_value \times in\_time\_price)}[Wallets \bowtie_{Market.market\_id=id} Markets]$$

۳.۲.۳ سوال ۳

$$_{crypto\_id}\mathcal{F}_{\texttt{Count}(order\_id)}[\sigma_{fill=" false"}(Orders)]$$

۳.۲.۴ سوال ۴

$$A = \rho_{user\_id, total}[_{owner\_id}\mathcal{F}_{\texttt{Sum}(fee) as total Sell}(Transactions \bowtie_{Transactions.origin\_wallet\_id=wallets.id} Wallet)]$$
$$B = \rho_{user\_id, total}[_{owner\_id}\mathcal{F}_{\texttt{Sum}(fee) as total Buy}(Transactions \bowtie_{Transactions.dest\_wallet\_id=wallets.id} Wallet)]$$

$$_{user\_id}\mathcal{F}_{\texttt{Sum}(Total)}$$

[۱] Relational Algebra

۳.۲.۵ سوال ۵

$$A =_{user\_id,cryptoid} \mathcal{F}_{\texttt{Count}(Transactions.id)}((Users \times Cryptocurrency) \ltimes_{users.user\_id=Transactions.SellerID} Transactions)$$

$$B =_{user\_id,cryptoid} \mathcal{F}_{\texttt{Count}(Transactions.id)}((Users \times Cryptocurrency) \ltimes_{users.user\_id=Transactions.BuyerID} Transactions)$$

$$_{user\_id,cryptoid}\mathcal{F}_{mathttSum(TotalCount)}(\rho_{user\_id,cryptoid/TotalCount(A)} \cup \rho_{user\_id,cryptoid/TotalCount(B)})$$

۳.۲.۶ سوال ۶

$$\mathcal{F}_{\texttt{Sum}(fee)}[\sigma_{Now-Date\geq\text{"}0000-00-30-00:00:00\text{"}}(Transactions)]$$

۳.۲.۷ سوال ۷

$$A = \Pi_{cryptoid,in\_time\_price}(Cryptocurrency)$$
$$B =_{cryptoid} \mathcal{F}_{mathttmax(Date)asDate}(\sigma_{Transactions.Date-Now()\leq\text{"}0000-00-30-00:00:00\text{"}}(Transactions \rtimes Cryptocurrency))$$
$$C = \Pi_{cryptoid,fee}[Cryptocurrency \bowtie_{Cryptocurrency.idTransactions.cryptoid} (Transactions \bowtie B)]$$

$$\Pi_{cryptoid,in\_time\_price-fee}(A \bowtie)C$$

۳.۲.۸ سوال ۸

$$A =_{owner\_id} \mathcal{F}_{\texttt{Sum}(Total\_value)assum}(Wallets)$$
$$B = \Pi_{owner\_id,cryptoid,Total\_value}(Wallets)$$
$$_{cryptoid}\mathcal{F}_{\texttt{count}(owner\_id)}[\sigma_{percentage\geq 0.05}(\rho_{cryptoid,owner\_id,percentage}[\Pi_{cryptoid,owner\_id,\frac{Total\_value}{Sum}}(A \bowtie B)])]$$

۳.۲.۹ سوال ۹

$$A = \Pi_{user\_id,Date}(\sigma_{Date-Now()\leq\text{"}0000-00-30;00:00:00\text{"}}(Online\_Payments))$$
$$B = \rho_{user\_id,paymentDate}(A) \bowtie WalletHistories$$
$$C =_{cryptoid,user\_id,paymentDate} \mathcal{F}_{\texttt{Max}(Date)}(\sigma_{Date<paymentDate}(B))$$
$$\rho_{user_id,paymentDate}(A) \times CryptoHistories$$
$$E =_{cryptoid,user\_id,paymentDate} \mathcal{F}_{\texttt{max}(Date)}(\sigma_{Date<paymentDate}(D))$$
$$X =_{user\_id,paymentDate} \mathcal{F}_{\texttt{Sum}(amount\times price)astotalValue}((C \times WalletHistories) \bowtie_{user\_id=user\_id\wedge paymentDate=paymentDate} E \bowtie CryptoHistories) \bowtie_{user\_id=user\_id\wedge paymentDate=paymentDate} Online\_Payments$$

$$\mathcal{F}_{\texttt{CountUnique(user\_id)}}(\sigma_{onlineamount\geq\frac{1}{5}totalValue}(X))$$

۳.۲.۱۰ سوال ۱۰

$$A = \rho_{cryptoid,price,totalSell}[_{cryptoid,price}\mathcal{F}_{\texttt{Sum(amount)}}((Cryptocurrency \times prices) \bowtie_{Cryptocurrency.id=sellOrders.cryptoid} sellOrders)]$$

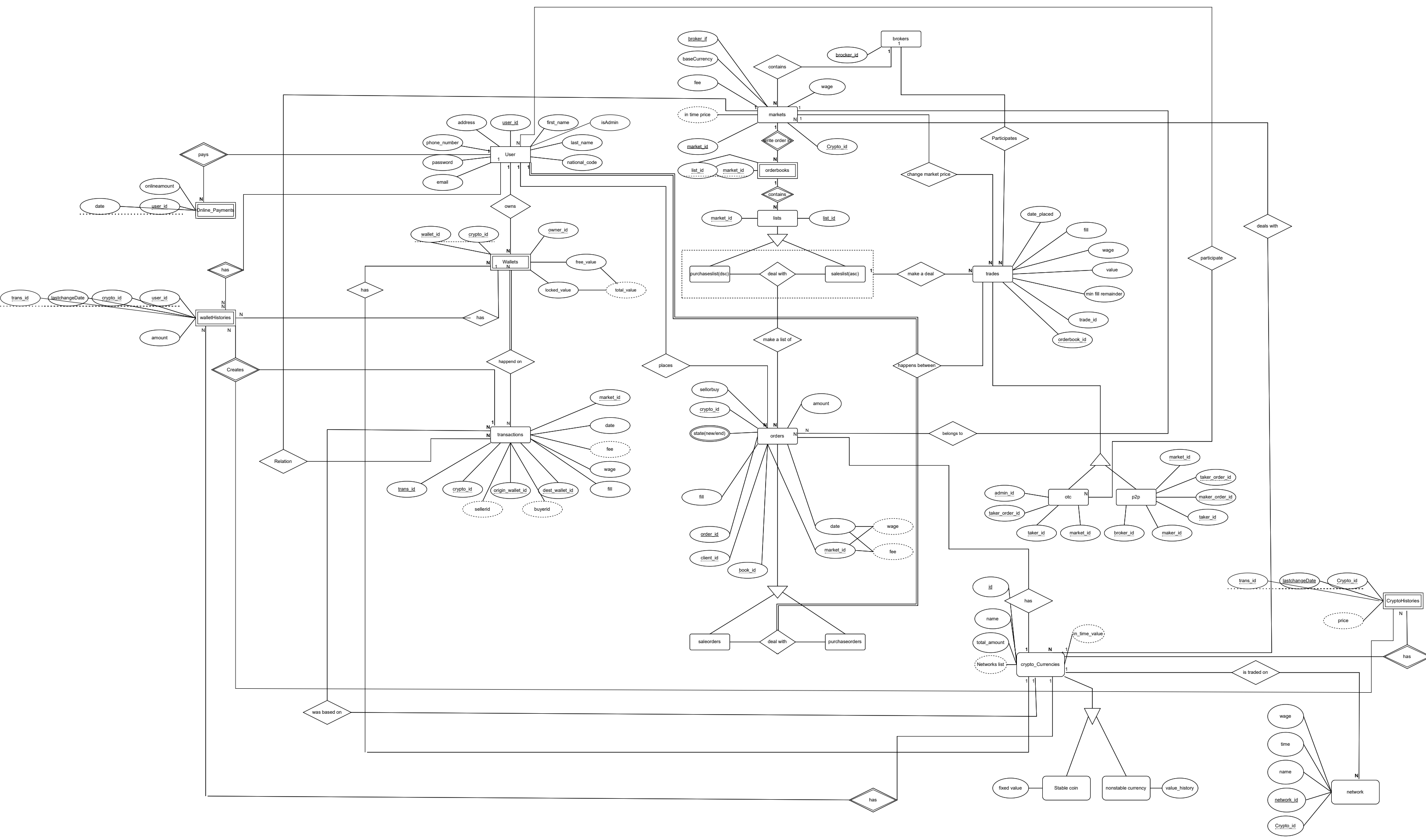
$$B = \rho_{cryptoid,price,totalSell}[_{cryptoid,price}\mathcal{F}_{\texttt{Sum(amount)}}((Cryptocurrency \times prices) \bowtie_{Cryptocurrency.id=purchaseOrders.cryptoid} purchaseOrders)]$$

$$A \cup B$$

فصل ۴

ضمیمه: تصویر دیاگرام ER

در انتهای فایل، ضمیمه‌ی تصویر دیاگرام مربوطه آمده است.
در صورتی که در مشاهده‌ی این تصویر مشکل دارید، فایل PDF را با مرورگرهای Edge یا Chrome باز نمایید.


brokers
broker_if
broker_id
baseCurrency
fee
contains
wage
markets
in time price
market_id
write order in
Crypto_id
orderbooks
list_id
contains
lists
change market price
Participates
deals with
participate
purchaseslist(dsc)
deal with
saleslist(asc)
make a deal
trades
date_placed
fill
value
min fill remainder
trade_id
orderbook_id
make a list of
happens between
places
belongs to
orders
sellorbuy
crypto_id
amount
state(new/end)
order_id
client_id
book_id
date
fee
saleorders
purchaseorders
otc
p2p
admin_id
taker_order_id
taker_id
market_id
broker_id
maker_id
maker_order_id
User
address
user_id
first_name
isAdmin
phone_number
last_name
password
national_code
email
pays
onlineamount
date
Online_Payments
owns
Wallets
wallet_id
owner_id
free_value
locked_value
total_value
has
walletHistories
trans_id
lastchangeDate
amount
Creates
happend on
transactions
origin_wallet_id
dest_wallet_id
sellerid
buyerid
Relation
was based on
id
name
total_amount
Networks list
in_time_value
crypto_Currencies
is traded on
Stable coin
nonstable currency
fixed value
value_history
network
time
network_id
Crypto_id
CryptoHistories
price

فصل ۵

کلیت فاز دوم پروژه

۵.۱ شرح

ما در این فاز از پروژه چهار تغیر در نموار فاز قبلی خود ایجاد کردیم که به ترتیب عبارتند از بهینه کردن نمودار برای طراحی دیتابیس، ایجاد دیتابیس، نرمال سازی و ایندکس کردن دیتا بیس و در نهایت انحام هشت جستوجو در دیتابیس.

فصل ۶

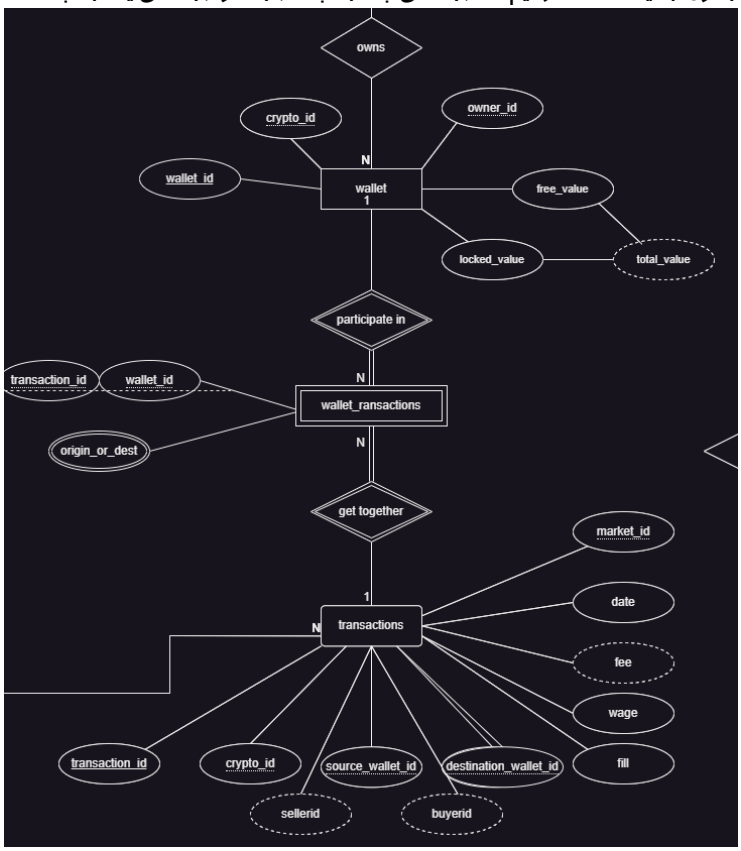
تبدیل نمودارهای فاز اول، به نمودارهای منطبق با SQL

۶.۱ چهار تغیر در نمودار برای بهینه سازی

در این بخش نمودار خود را ۴غیر دادیم تا مناسب درست کردن SQL باشد.

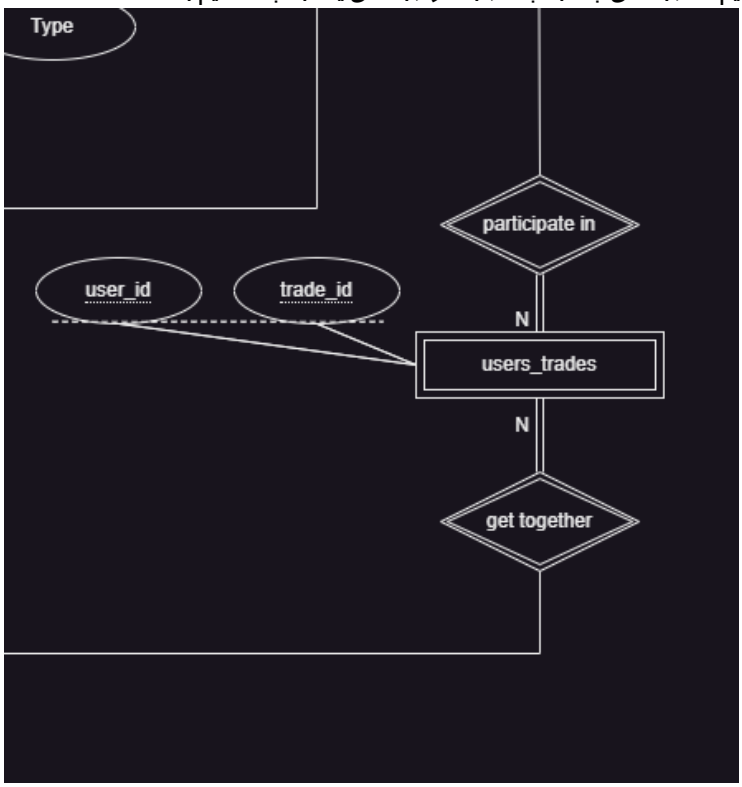
۶.۱.۱ رابطه ی چند به چند بین کیف پول و تراکنش‌ها

از انحایی که در هر تراکنش دو کیف پول استفاده میشد و هر کیف پول در چندین تراکنش شرکت میکرد، یک جدول جدید اضافه کردیم که رابطه ی چند به چند را به دو رابطه ی یک به چند تقسیم کند.



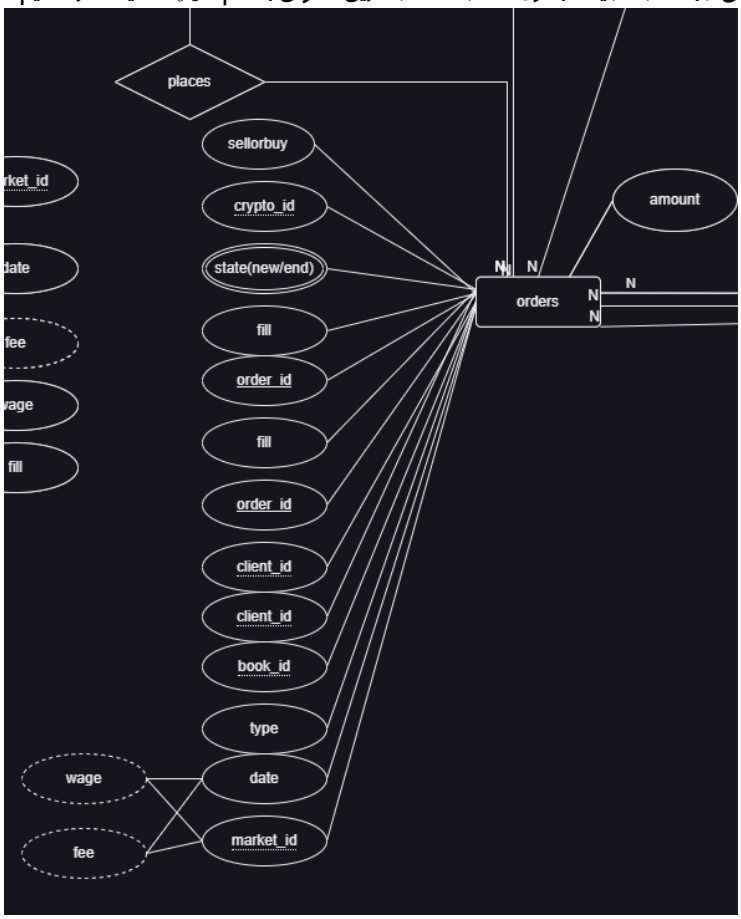
۶.۱.۲ رابطه ی چند به چند بین کاربرها و تبادل ها

از انجایی که هر تبادل از دو کاربر و هر کاربر در چندین تبادل شرکت میکرد، یک جدول جدول جدید اضافه کردیم که رابطه ی چند به چند را به دو رابطه ی یک به چند تسیم بکمد.



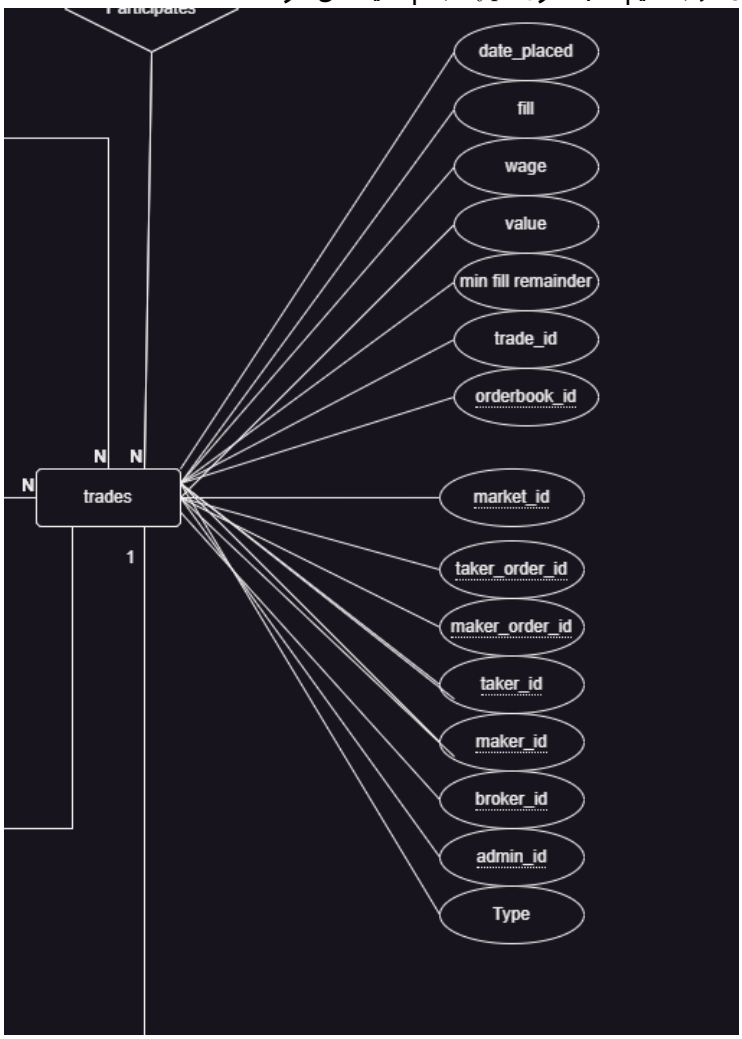
۶.۱.۳ رابطه ی اسپیسیفیکیشن در سفارشات

در فاز قبلی سفارشات به دو دسته‌ی سفارشت خرید و سفارشات فروش تقسیم می‌شدند که ما در این فاز ای رابطه را در یک جدول سفارشات از طریق ستونی به نام type تفکیک کرده ایم.



۶.۱.۴ رابطه ی جنرالیزیشن در تبادل‌ها

در فاز قبلی تبادل‌ها شامل دو نوع $\mathrm{p2p}$ و otc می‌شدند که ما در این فاز آن دو نوع تبادل را در یک جدول تبادل قرار دادیم که با ستون type از هم تفکیک می شوند.



فصل ۷

ساخت پایگاه داده

بر اساس نمودار بخش قبل دو فایل SQL قرار دادیم که در یکی دستورات ایجاد جدول‌ها و دیگری تست کیس برای هر جدول ایجاد شده و در مسیر Phase2/SQL Files قرار داده شده است.

فصل ۸

بهبود پایگاه داده

۸.۱ نرمال‌تر سازی

در این بخش ما تمام جداول پایگاه داده‌ی خود را به فرم نرمال در اوردیم و تغیرات نمودار و دستورات ایجاد پایگاه داده را در دو فایل normalized sqlform Integrated.drawio **و** Normalized Cre-ate Tables.sql **در مسیر** Phase2/Normalize Files **قرار دادیم.**

۸.۱.۱ حذف crypto__id از transactions

$$transactions(\underline{transaction\_id}, crypto\_id, source\_wallet\_id, destination\_wallet\_id, fill, wage, date, market$$

$$F.D = \{transaction\_id \rightarrow allattributes, market\_id \rightarrow crypto\_id\}$$

از انحایی که در $market\_id \rightarrow crypto\_id$ **یک** non prime attribute **به یک** non prime attribute **دیگر اشاره کرده این دیپندنسی را باید در یک جدول دیگر قرار دهیم تا از دومین فرم نرمال به سومین فرم نرمال انتقال پیدا کنیم.**

$$transactions(transaction\_id, source\_wallet\_id, destination\_wallet\_id, fill, wage, date, market\_id)$$

$$F.D = \{transaction\_id \rightarrow all\_attributes\}$$

$$R(\underline{market\_id}, crypto\_id)$$

$$F.D = \{market\_id \rightarrow crypto\_id\}$$

حال دو جدول ما دارای سومین فرم نرمال هستند که همانطور که میبینید جدول R **زیر مجموعه ای از جدول** Markets **در پایگاه داده اصلی میباشد و نیازی به ساختن جدول اضافه نیست.**

۸.۱.۲ حذف list__id ها از orders

۸.۱.۳ حذف maker__id, taker__id از trades

۸.۲ indexها

برای سه جدول خود index **قرار دادیم تا در جستار ها به ما کمک بکند.**

۸.۲.۱ ایندکس تاریخ بر روی تبادل‌ها

از انحایی که ما نیاز داریم تا قیمت لحظه‌ای هر بازار را بر اساس اخرین تبادل ثبت شده حساب کنیم. مرتب کردن این حدول بر اساس تاریخ به محاسبه‌ی قیمت بازار بسیار کمک می‌کند.

```
CREATE INDEX IF NOT EXISTS date_placed_idx ON trades(date_placed);
```

۸.۲.۲ ایندکس کیف پول بر روی تراکنش‌ها

از انحایی که ما نیازمند محاسبه‌ی موجودی کیف پول هر کاربر هستیم پس بهتر است تراکنش های هر کیف پول را مرتب و درکنار هم در جدول تبادل ها قرار دهیم.

```
CREATE INDEX IF NOT EXISTS walle_id_idx
    ON transactions(source_wallet_id);
```

۸.۲.۳ ایندکس قیمت بر روی سفارش‌ها

از انحایی که تر تبادل کم قیمت‌ترین سفارش فروش مورد استفاده قرار می‌گیرد بهتر است که بر اساس قیمت سفارشات خود را مرتب کنیم تا در هر لحظه دسترسی سریعی به کم قیمت‌ترین پیشنهاد فروش داشته باشیم.

```
CREATE INDEX IF NOT EXISTS amount_idx ON orders(amount);
```