

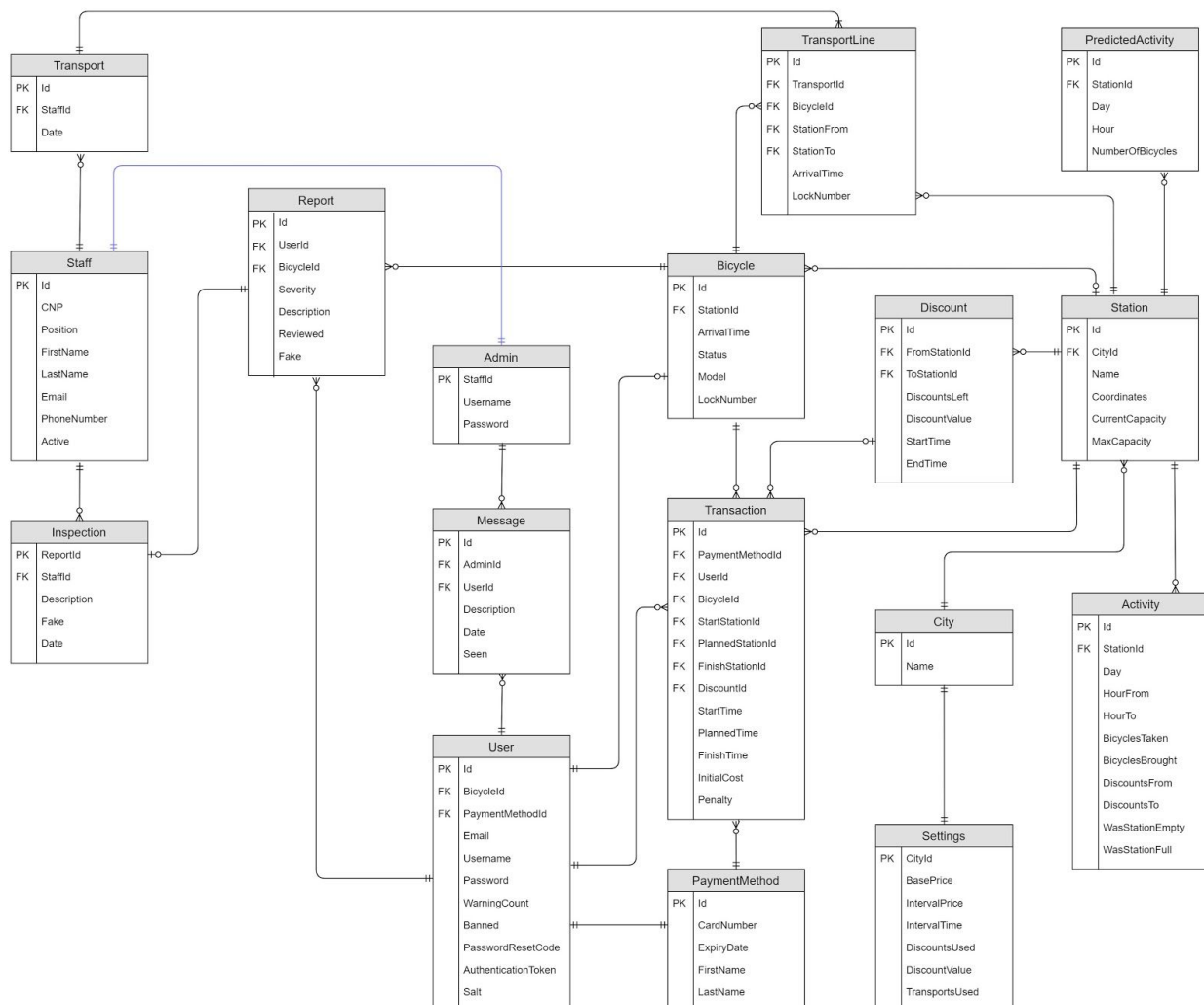
# Licență 2020 Facultatea de Informatică UAIC

## Aplicație pentru partajarea utilizării bicicletelor

Baza de date

Cocei Tiberiu

### 1. Schema bazei de date



## 2. Explicația tabelelor

### 1. User

Tabela această va avea lista de conturi ale utilizatorilor obișnuiți. Va avea o cheie străină pentru tabelele PaymentMethod și Bicycle. Cheia pentru tabela Bicycle va fi null dacă user-ul nu are o bicicletă la momentul respectiv.

Tabela va mai reține e-mail-ul, username-ul și parola utilizatorilor. Parola va fi criptată folosind o funcție hash. Coloana Salt va reține în plain text salt-ul utilizatorului pentru funcția hash.

Coloana WarningCount va reține numărul de avertismente ale utilizatorilor. Odată ce ajunge la un anumit număr, utilizatorul își va pierde dreptul de acces la aplicație, atributul Banned devenind true.

Coloana PasswordResetCode va reține codul pentru a reseta parola. Ca valoare de default va fi null, memorând un text generat random la cererea utilizatorului. Coloana AuthenticationToken va reține un UUID generat random la login care va fi cerut de API-uri.

### 2. Bicycle

Tabela această va avea lista de biciclete accesibile prin intermediul aplicației. Va avea o cheie străină pentru tabela Station, ce reprezintă stația în care se află bicicleta. Dacă bicicleta este în uz, va reprezenta stația la care va ajunge.

Atributul Status va avea o enumerație pentru stări referitoare la bicicletă: Station, Warehouse, Transport, User, Damaged și Stolen. Dacă acest atribut nu are valoarea Station, nu va fi valabil pentru utilizatori.

Dacă nu este în folosință, atributul ArrivalTime va fi null. Altfel, atributul va avea o aproximare pentru ora la care va ajunge în stație.

Coloana LockNumber va avea numărul lacătului din stația în care este pusă bicicleta. Dacă este în folosință sau în depozit, atributul va fi null.

### **3. Station**

Tabela această va avea lista de stații pentru biciclete. Va conține orașul și coordonatele unde este situată precum și numărul curent și maxim de biciclete.

### **4. Transaction**

Tabela această va avea lista de tranzacții efectuate. Va avea chei străine pentru tabelele User, Bicycle, Payment, Station și Discount.

Va conține chei străine către stația de unde utilizatorul a luat o bicicletă, unde a spus că va duce bicicleta și unde a dus-o în realitate. De asemenea, va conține timpul de plecare, timpul când utilizatorul a spus că va ajunge și timpul când a ajuns.

Coloana InitialCost va conține costul pentru estimarea inițială și Penalty va conține costul adăugat. Penalty poate fi adăugat din mai multe motive, cum ar fi întârzierea de a ajunge la stație, schimbarea stației sau furtul bicicletei. Penalty-ul va fi dedus de pe cardul de credit al utilizatorului.

Discount reprezintă valoarea ce a fost dedusă din InitialCost deoarece există o reducere pentru plecarea din stația X și ajungerea în stația Y. Dacă stația unde este dusă bicicletă diferă sau dacă utilizatorul întârzie cu prea mult timp, valoarea va fi adăugată în întregime la atributul Penalty.

### **5. PaymentMethod**

Tabela această va avea lista cu detalii de plată ale fiecărui utilizator.

## **6. Discount**

Tabela această va avea lista cu reduceri între stații. Va avea o cheie străină către stația de plecare și una către stația de sosire. De asemenea, va avea numărul de reduceri rămase, valoarea reducerii și timpul de start și de final a reducerii.

Aceste reduceri pot fi create manual de un administrator sau generate automat de un algoritm.

## **7. Report**

Tabela această va avea lista cu rapoarte făcute de utilizatori. Va avea chei străine pentru tabelele User și Bicycle. De asemenea, va avea un grad de severitate și o descriere.

Atributul boolean Reviewed, inițial false, va deveni true odată ce un angajat verifică bicicleta. Atributul boolean Fake va fi inițial null. Odată ce a fost verificată bicicleta, va deveni false sau true, dacă raportul e fals respectiv adevărat. Dacă raportul este fals și severitatea raportului mare, user-ul va primi un avertisment.

## **8. Inspection**

Tabela această va avea lista cu inspecții făcute de angajați. Va avea chei străine pentru tabelele Staff și Report, cea pentru Report fiind și cheia primară. De asemenea, va avea o descriere, data în care a fost efectuată inspecția și dacă raportul este adevărat.

## **9. Message**

Tabela această va avea lista cu mesaje ale utilizatorilor făcute de administratori. Va avea chei străine pentru tabelele Admin și User. De asemenea, va conține textul mesajului, data la care a fost trimis și un atribut boolean dacă a fost văzut de utilizator.

## **10. Staff**

Tabela această va avea lista cu angajați. Va conține detalii personale și tipul de muncă efectuat.

## **11. Admin**

Tabela această va avea lista de administratori. Va conține username-ul și parola criptată cu o funcție hash. Restul datelor vor putea fi preluate printr-un join cu tabela Staff, cheia primară fiind și una străină.

## **12. Activity**

Tabela această va avea activitatea zilnică pentru fiecare stație pentru anumite intervale de timp. O constrângere de unicitate va fi formată din trei atribute: Id-ul stației, ziua și ora de început, astfel încât acest tuplu trebuie să fie unic. Ora de final va fi ora de început +1.

Coloanele BicyclesTaken și BicyclesBrought vor memora numărul de biciclete luate din stație respectiv aduse. Coloanele DiscountsFrom și DiscountsTo vor memora numărul de reduceri pentru plecarea și pentru ajungerea în stație. Atributele boolean-e WasStationEmpty și WasStationFull vor memora dacă stația a fost goală respectiv plină în intervalul de timp dat.

Informațiile din acest tabel vor fi folosite de un algoritm pentru a genera predicții pentru numărul de biciclete ce ar trebui să fie într-o stație specifică la o anumită oră.

## **13. PredictedActivity**

Tabela această va avea predicțiile generate de un algoritm pentru numărul de biciclete ce ar trebui să fie într-o stație la o anumită oră.

O constrângere de unicitate va fi formată din trei atribute: Id-ul stației, ziua și ora, astfel încât acest tuplu trebuie să fie unic. Atributul NumberOfBicycles reprezintă numărul de biciclete generat de algoritm.

## **14. Settings**

Tabela această va avea setările folosite pentru aplicație. Cheia primară va fi id-ul orașului pentru care sunt menite setările, relația fiind de 1:1.

Coloana BasePrice va memora prețul de start al unui drum cu o bicicletă și coloana IntervalPrice costul care va fi adăugat pentru fiecare IntervalTime minute.

Boolean-ele DiscountsUsed și TransportsUsed vor influența algoritmul de partajare al bicicletelor și DiscountValue va reprezenta valoarea de bază a unei reduceri.

## **15. Transport**

Tabela această va avea lista cu transporturile de biciclete cu un autovehicul efectuate. Va avea o cheie străină pentru tabela Staff și data efectuării transportului.

## **16. TransportLine**

Tabela această va avea lista cu biciclete transportate. Va avea chei străine pentru tabela Transport: transportul din care face parte, Bicycle: Id-ul bicicletei și Station: stația de unde a fost luată bicicleta și unde a fost dusă. Atributul LockNumber va memora numărul lacătului unde a fost pusă bicicleta. Atributul ArrivalTime va avea o aproximare pentru când va ajunge la destinație.

## **17. City**

Tabela această va avea două atribute: Id-ul și numele orașului.

### 3. Triggere, funcții și reguli

Regulile sunt simple: previn operațiile de ștergere asupra oricărei tabele și de updatare asupra tabelelor Transport, Transport\_line, Inspection, City și Predicted\_activity.

Există nouă triggere, fiecare având câte o funcție pe care o apelează:

1. **transport\_line\_update\_on\_station\_and\_bicycle:** după ce este inserată o linie în tabela TransportLine, tabelele Station și Bicycle vor fi updatate. În Station stațiile de plecare și de sosire vor avea current\_capacity schimbat (decrementat pentru stația de plecare, incrementat pentru stația de sosire), iar în Bicycle va fi schimbat status (devine 'Transport') și arrival\_time (devine timpul estimat de sosire).
2. **inspection\_update\_on\_report:** după ce este inserată o linie în tabela Inspection, tabela Report va fi updatată. Coloanele review și fake vor fi schimbate (sunt preluate din Inspection).
3. **report\_update\_on\_user\_and\_bicycle:** după un update pe tabela Report, în funcție de nivelul de severitate și dacă nu este fals reportul, statusul bicicletei va deveni 'Damaged' și nu va mai putea fi folosită, urmând să fie luată din stație. Dacă reportul este fals, user-ul va primi un avertisment.
4. **user\_on\_update\_check\_for\_ban:** după update pe coloana warning\_count în tabela User, va fi verificat numărul de avertismente. Dacă este mai mare sau egal decât trei, user-ul își va avea accesul la aplicație restricționat.
5. **discount\_update\_on\_activity:** după ce este inserată o linie în tabela Discount, tabela Activity va fi updatată. Pentru tabela de plecare coloana discounts\_from și pentru tabela de sosire coloana discounts\_to vor fi actualizate, fiind incrementate cu numărul total de reduceri.

6. **station\_on\_update\_change\_activity:** după un update pe tabela Station, dacă stația rămâne fără biciclete sau devine plină atunci coloanele was\_station\_empty respectiv was\_station\_full vor fi updatate în tabela Activity (vor deveni True).
7. **transaction\_on\_insert\_modifies\_tables:** după ce este inserată o linie în tabela Transaction, mai multe tabele vor fi updatate. Tabela bicycle își va avea stația și timpul de sosire schimbat în funcție de alegerea user-ului. Tabela station va avea current\_capacity schimbat atât pentru stația de plecare cât și de sosire. Tabela user va avea id-ul bicicletei schimbat, astfel încât user-ul să nu poată avea mai multe biciclete simultan. Dacă un discount a fost folosit, tabela Discount va fi modificată astfel încât numărul să fie decrementat. Tabela Activity va fi schimbată, fiind incrementat numărul de biciclete luate din stația de plecare.
8. **transaction\_on\_update\_modifies\_activity\_and\_penalty:** după ce a fost updatată o linie în tabela Transaction, vor fi calculate penalitățile user-ului pentru cazul în care a întârziat sau a dus bicicleta în stația greșită. În tabela User user-ul va avea id-ul bicicletei setat la NULL, în tabela Activity numărul de biciclete aduse va fi incrementat pentru stația de sosire și dacă stația de sosire este alta față de cea aleasă de utilizator, acesta va primi un avertisment și coloanele de current\_capacity vor fi schimbate pentru stația aleasă original și cea de sosire.
9. **user\_on\_update\_create\_message:** după update pe coloana warning\_count în tabela User, un mesaj va fi creat pentru user, notificându-l de avertisment.



## 4. Indecși

Există 14 indecși diferiți, toți fiind de tip B-tree. În PostgreSQL, indecșii de tip Bitmap nu există și cei de tip Hash prezintă anumite probleme de utilizare (în cazul unui crash).

1. transport (date): Index pentru statistici legate de transporturi.
2. transport\_line (transport\_id): Index pentru transportul din care face parte linia respectivă.
3. report (bicycle\_id, reviewed): Index pentru reporturile fiecărei biciclete.
4. app\_admin (username): Index pentru logarea administratorilor.
5. message (user\_id): Index pentru preluarea mesajelor utilizatorilor.
6. app\_user (username): Index pentru logarea utilizatorilor.
7. bicycle (station\_id): Index pentru preluarea bicicletelor din stații.
8. app\_transaction (start\_time): Index pentru statistici legate de tranzacții de-a lungul timpului.
9. app\_transaction (start\_station\_id, start\_time): Index pentru statistici legate de tranzacții de-a lungul timpului pentru plecarea din stații specifice.
10. app\_transaction (finish\_station\_id, start\_time): Index pentru statistici legate de tranzacții de-a lungul timpului pentru sosirea în stații specifice.
11. station (city\_id): Index pentru preluarea stațiilor dintr-un oraș.
12. predicted\_activity (station\_id, day, hour): Index pentru preluarea activității prevăzute a unei stații pentru o zi și ora specifică.
13. discount (from\_station\_id, to\_station\_id, start\_time): Index pentru preluarea reducerilor de la o stație la alta la o oră specifică.
14. activity (station\_id, day, hour\_from): Index pentru preluarea activității unei stații pentru o zi și oră specifică.

## 5. Script de generare date

Un script a fost creat folosind limbajul de programare Python pentru a popula majoritatea tabelelor din baza de date. Tabelele Inspection, Report, Message, Discount și PredictedActivity au rămas goale astfel încât popularea lor cu date realiste ar fi fost destul de complicată și nu este necesară pentru utilizarea aplicației sau pentru algoritmul de prezicere a activității stațiilor.

Tabelele City, Settings și Station nu sunt generate aleator, astfel încât să aibă cât mai mult sens. Tabelele User, PaymentMethod, Admin, Bicycle și Staff au fost generate în mod aleator, folosind un API extern pentru a popula o listă de nume și de prenume din care au fost alese.

Începând cu 1 ianuarie până la 30 iunie, au fost generate în mod aleator tabelele Transaction, Activity, Transport, TransportLine. Luând oră cu oră, de la 7 la 21, au fost generate tranzacții pentru fiecare stație în parte. Dacă o stație era aproape de a fi goală sau plină, atunci erau create transporturi pentru a remedia situația.