

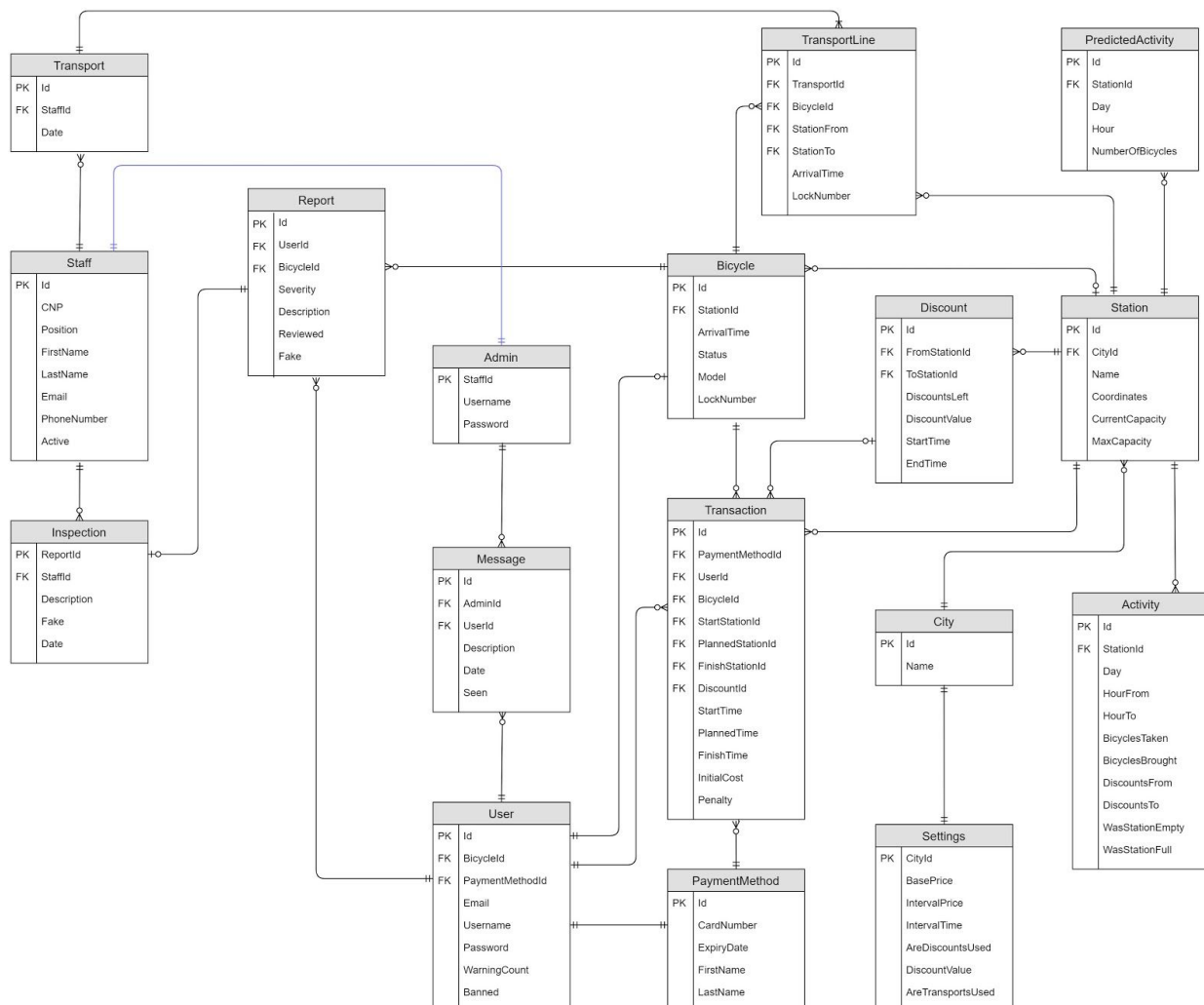
# Licență 2020 Facultatea de Informatică UAIC

## Aplicatie pentru partajarea utilizarii bicicletelor

Baza de date

Cocei Tiberiu

### 1. Schema bazei de date



## **2. Explicatia tabelelor**

### **1. User**

Tabela aceasta va avea lista de conturi ale utilizatorilor obisnuiti. Va avea o cheie straina pentru tabelele PaymentMethod si Bicycle. Cheia pentru tabela Bicycle va fi null daca user-ul nu are o bicicleta la momentul respectiv.

Tabela va mai retine e-mail-ul, username-ul si parola utilizatorilor. Parola va fi criptata folosind o functie hash.

Coloana WarningCount va retine numarul de avertismente ale utilizatorilor. Odata ce ajunge la un anumit numar, utilizatorul isi va pierde dreptul de acces la aplicatie, atributul boolean Banned devenind true.

### **2. Bicycle**

Tabela aceasta va avea lista de biciclete accesibile prin intermediul aplicatiei. Va avea o cheie straina pentru tabela Station, ce reprezinta statia in care se afla bicicleta. Daca bicicleta este in uz, va reprezenta statia la care va ajunge.

Atributul Status va avea o enumeratie pentru stari referitoare la bicicleta: Station, Warehouse, Transport, User, Damaged si Stolen. Daca acest atribut nu are valoarea Station, nu va fi valabil pentru utilizatori.

Daca nu este in folosinta, atributul ArrivalTime va fi null. Altfel, atributul va avea o aproximare pentru ora la care va ajunge in statie.

Coloana LockNumber va avea numarul lacatului din statia in care este pusa bicicleta. Daca este in folosinta sau in depozit, atributul va fi null.

### **3. Station**

Tabela aceasta va avea lista de statii pentru biciclete. Va contine orasul si coordonatele unde este situata precum si numarul curent si maxim de biciclete.

### **4. Transaction**

Tabela aceasta va avea lista de tranzactii efectuate. Va avea chei straine pentru tabelele User, Bicycle, Payment, Station si Discount.

Va contine chei straine catre statia de unde utilizatorul a luat o bicicleta, unde a spus ca va duce bicicleta si unde a dus-o in realitate. De asemenea, va contine timpul de plecare, timpul cand utilizatorul a spus ca va ajunge si timpul cand a ajuns.

Coloana InitialCost va contine costul pentru estimarea initiala si Penalty va contine costul adaugat. Penalty poate fi adaugat din mai multe motive, cum ar fi intarzierea de a ajunge la statie, schimbarea statiei sau furtul bicicletei. Penalty-ul va fi dedus de pe cardul de credit al utilizatorului.

Discount reprezinta valoarea ce a fost dedusa din InitialCost deoarece exista o reducere pentru plecarea din statia X si ajungerea in statia Y. Daca statia unde este dusa bicicleta difera sau daca utilizatorul intarzie cu prea mult timp, valoarea va fi adaugata in intregime la atributul Penalty.

### **5. PaymentMethod**

Tabela aceasta va avea lista cu detalii de plata ale fiecarui utilizator.

## **6. Discount**

Tabela aceasta va avea lista cu reduceri intre statii. Va avea o cheie straina catre statia de plecare si una catre statia de sosire. De asemenea, va avea numarul de reduceri ramase, valoarea reducerii si timpul de start si de final a reducerii.

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

## **7. Report**

Tabela aceasta va avea lista cu rapoarte facute de utilizatori. Va avea chei straine pentru tabelele User si Bicycle. De asemenea, va avea un grad de severitate si o descriere.

Atributul boolean Reviewed, initial false, va deveni true odata ce un angajat verifica bicicleta. Atributul boolean Fake va fi initial null. Odata ce a fost verificata bicicleta, va deveni false sau true, daca raportul e fals respectiv adevarat. Daca raportul este fals si severitatea raportului mare, user-ul va primi un avertisment.

## **8. Inspection**

Tabela aceasta va avea lista cu inspectii facute de angajati. Va avea chei straine pentru tabelele Staff si Report, cea pentru Report fiind si cheia primara. De asemenea, va avea o descriere, data in care a fost efectuata inspectia si daca raportul este adevarat.

## **9. Message**

Tabela aceasta va avea lista cu mesaje ale utilizatorilor facute de administratori. Va avea chei straine pentru tabelele Admin si User. De asemenea, va contine textul mesajului, data la care a fost trimis si un atribut boolean daca a fost vazut de utilizator.

## **10. Staff**

Tabela aceasta va avea lista cu angajati care nu sunt administratori. Va contine detalii personale si tipul de munca efectuat.

## **11. Admin**

Tabela aceasta va avea lista de administratori. Va contine username-ul si parola criptata cu o functie hash. Restul datelor vor putea fi preluate printr-un join cu tabela Staff, cheia primara fiind si una straina.

## **12. Activity**

Tabela aceasta va avea activitatea zilnica pentru fiecare statie pentru anumite intervale de timp. O constrangere de unicitate va fi formata din trei attribute: Id-ul statiei, ziua si ora de inceput, astfel incat acest tuplu trebuie sa fie unic. Ora de final va fi ora de inceput +1.

Coloanele BicyclesTaken si BicyclesBrought vor memora numarul de biciclete luate din statie respectiv aduse. Coloanele DiscountsFrom si DiscountsTo vor memora numarul de reduceri pentru plecarea si pentru ajungerea in statie. Attributele boolean-e WasStationEmpty si WasStationFull vor memora daca statia a fost goala respectiv plina in intervalul de timp dat.

Informatiile din acest tabel vor fi folosite de un algoritm pentru a genera predictii pentru numarul de biciclete ce ar trebui sa fie intr-o statie specifica la o anumita ora.

## **13. PredictedActivity**

Tabela aceasta va avea predictiile generate de un algoritm pentru numarul de biciclete ce ar trebui sa fie intr-o statie la o anumita ora.

O constrangere de unicitate va fi formata din trei attribute: Id-ul statiei, ziua si ora, astfel incat acest tuplu trebuie sa fie unic. Atributul NumberOfBicycles reprezinta numarul de biciclete generat de algoritm.

## **14. Settings**

Tabela aceasta va avea setarile folosite pentru aplicatie. Cheia primara va fi id-ul orasului pentru care sunt menite setarile, relatia fiind de 1:1.

Coloana BasePrice va memora pretul de start al unui drum cu o bicicleta si coloana IntervalPrice costul care va fi adaugat pentru fiecare IntervalTime minute.

Boolean-ele AreDiscountsUsed si AreTransportsUsed vor influenta algoritmul de partajare al bicicletelor si DiscountValue va reprezenta valoarea de baza a unei reduceri.

## **15. Transport**

Tabela aceasta va avea lista cu transporturile de biciclete cu un autovehicul efectuate. Va avea o cheie straina pentru tabela Staff si data efectuarii transportului.

## **16. TransportLine**

Tabela aceasta va avea lista cu biciclete transportate. Va avea chei straine pentru tabela Transport: transportul din care face parte, Bicycle: Id-ul bicicletei si Station: statia de unde a fost luata bicicleta si unde a fost dusa. Atributul LockNumber va memora numarul lacatului unde a fost pusa bicicleta. Atributul ArrivalTime va avea o aproximare pentru cand va ajunge la destinatie.

## **17. City**

Tabela aceasta va avea doua atribute: Id-ul si numele orasului.

### 3. Triggere, functii si reguli

Regulile sunt simple: previn operatiile de stergere asupra oricarei tabele si de updatare asupra tabelelor Transport, Transport\_line, Inspection, City si Predicted\_activity.

Exista noua triggere, fiecare avand cate o functie pe care o apeleaza:

- 1. transport\_line\_update\_on\_station\_and\_bicycle:** dupa ce este inserata o linie in tabela TransportLine, tabelele Station si Bicycle vor fi updatate. In Station statiile de plecare si de sosire vor avea current\_capacity schimbat (decrementat pentru statia de plecare, incrementat pentru statia de sosire), iar in Bicycle va fi schimbat status (devine 'Transport') si arrival\_time (devine timpul estimat de sosire).
- 2. inspection\_update\_on\_report:** dupa ce este inserata o linie in tabela Inspection, tabela Report va fi updatata. Coloanele review si fake vor fi schimbate (sunt preluate din Inspection).
- 3. report\_update\_on\_user\_and\_bicycle:** dupa un update pe tabela Report, in functie de nivelul de severitate si daca nu este fals reportul, statusul bicicletei va deveni 'Damaged' si nu va mai putea fi folosita, urmand sa fie luata din statie. Daca reportul este fals, user-ul va primi un avertisment.
- 4. user\_on\_update\_check\_for\_ban:** dupa update pe coloana warning\_count in tabela User, va fi verificat numarul de avertismente. Daca este mai mare sau egal decat trei, user-ul isi va avea accesul la aplicatie restrictionat.
- 5. discount\_update\_on\_activity:** dupa ce este inserata o linie in tabela Discount, tabela Activity va fi updatata. Pentru tabela de plecare coloana discounts\_from si pentru tabela de sosire coloana discounts\_to vor fi actualizate, fiind incrementate cu numarul total de reduceri.

6. **station\_on\_update\_change\_activity:** dupa un update pe tabela Station, daca statia ramane fara biciclete sau devine plina atunci coloanele was\_station\_empty respectiv was\_station\_full vor fi updatate in tabela Activity (vor deveni True).
7. **transaction\_on\_insert\_modifies\_tables:** dupa ce este inserata o linie in tabela Transaction, mai multe tabele vor fi updatate. Tabela bicycle isi va avea statia si timpul de sosire schimbat in functie de alegerea user-ului. Tabela station va avea current\_capacity schimbat atat pentru statia de plecare cat si de sosire. Tabela user va avea id-ul bicicletei schimbat, astfel incat user-ul sa nu poata avea mai multe biciclete simultan. Daca un discount a fost folosit, tabela Discount va fi modificata astfel incat numarul sa fie decrementat. Tabela Activity va fi schimbata, fiind incrementat numarul de biciclete luate din statia de plecare.
8. **transaction\_on\_update\_modifies\_activity\_and\_penalty:** dupa ce a fost updatata o linie in tabela Transaction, vor fi calculate penalitatile user-ului pentru cazul in care a intarziat sau a dus bicicleta in statia gresita. In tabela User user-ul va avea id-ul bicicletei setat la NULL, in tabela Activity numarul de biciclete aduse va fi incrementat pentru statia de sosire si daca statia de sosire este alta fata de cea aleasa de utilizator, acesta va primi un avertisment si coloanele de current\_capacity vor fi schimbate pentru statia aleasa original si cea de sosire.
9. **user\_on\_update\_create\_message:** dupa update pe coloana warning\_count in tabela User, un mesaj va fi creat pentru user, notificandu-l de avertisment.



## 4. Indecsi

Exista 14 indecsi diferiti, toti fiind de tip B-tree. In PostgreSQL, indecsii de tip Bitmap nu exista si cei de tip Hash prezinta anumite probleme de utilizare (in 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 respectiva.
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 statii.
8. app\_transaction (start\_time): Index pentru statistici legate de tranzactii de-a lungul timpului.
9. app\_transaction (start\_station\_id, start\_time): Index pentru statistici legate de tranzactii de-a lungul timpului pentru plecarea din statii specifice.
10. app\_transaction (finish\_station\_id, start\_time): Index pentru statistici legate de tranzactii de-a lungul timpului pentru sosirea in statii specifice.
11. station (city\_id): Index pentru preluarea statiilor dintr-un oras.
12. predicted\_activity (station\_id, day, hour): Index pentru preluarea activitatii prevazute a unei statii pentru o zi si ora specifica.
13. discount (from\_station\_id, to\_station\_id, start\_time): Index pentru preluarea reducerilor de la o statie la alta la o ora specifica.
14. activity (station\_id, day, hour\_from): Index pentru preluarea activitatii unei statii pentru o zi si ora specifica.

## 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 si PredictedActivity au ramas goale astfel incat popularea lor cu date realiste ar fi fost destul de complicata si nu este necesara pentru utilizarea aplicatiei sau pentru algoritmul de prezicere a activitatii statiilor.

Tabelele City, Settings si Station nu sunt generate aleator, astfel incat sa aiba cat mai mult sens. Tabelele User, PaymentMethod, Admin, Bicycle si Staff au fost generate in mod aleator, folosind un API extern pentru a popula o lista de nume si de prenume din care au fost alese.

Incepand cu 1 ianuarie pana la 30 iunie, au fost generate in mod aleator tabelele Transaction, Activity, Transport, TransportLine. Luand ora cu ora, de la 7 la 21, au fost generate tranzactii pentru fiecare statie in parte. Daca o statie era aproape de a fi goala sau plina, atunci erau create transporturi pentru a remedia situatia.

Informatiile necesare pentru conectarea la baza de date sunt stocate in fisierul database.ini. Acestea nu sunt criptate, astfel incat scriptul face o simulare pentru activitatea zilnica a aplicatiei intr-o perioada de timp, ceva ce nu ar trebui folosit intr-o aplicatie reala. Astfel, nu prezinta un risc din punct de vedere al securitatii.