



REPUBLIKA E SHQIPËRISË
UNIVERSITETI I TIRANËS
FAKULTETI I SHKENCAVE TË NATYRËS

Lënda: Databaza nën Oracle

Semestri I

Tema: Sistem për menaxhimin e të dhënave të një metroje

Punuan:

[Student1] Albi Ndroqi
[Student2] Albi Taulla
[Student3] Erska Doka
[Student4] Iris Dizdari

Pranoi:

Dr. Silvana Greca

Përmbajtja:

	Grupi	1
1.	Përshkrimi i bazës së të dhënave	3
2.	Skema ER/EER	4
3.	Relacionet	5
4.	Krijimi i bazës së të dhënave	7
5.	Query	11
6.	View	20
7.	Indexet	23

PËRSHKRIMI I BAZËS SË TË DHËNAVE

Databaza është ndertuar mbi idene menaxhimit të të dhënave të një sistemi metroje. Databaza ka si qëllim të mundësojë funksionalitete si marrja e informacionit për trenat e disponueshëm bazuar mes dy stacioneve, blerja e biletave dhe anulimi i tyre, marrja e informacionit mbi një biletë, etc. Databaza lejon rezervimin e një bilete duke marrë informacion mbi Idnë e trenit dhe udhëtimin që ky tren realizon në një orar të caktuar. Gjithashtu databaza ruan informacion mbi të gjithë stacionet e kësaj metroje, punonjësit ndryshëm që ajo mund të ketë dhe rrugët që lidhin këto stacione.

Entiteti: Person, ruan informacion mbi një person që ne rastin tonë mund të jete një punonjes ose një pasagjer i thjeshtë. Informacionet janë të dhëna bazike të një personi në shoqëri,

Atributet:

ID - id identifikuese e personit të zakonshëm

Emër – emri i personit

Mbiemër - mbiemri i personit

Gjinia – gjinia e personit

Data_E_Lindjes – datëlindja e personit

Email – email i personit

Shteti- shteti ku jeton ky person

Qyteti – qyteti ku jeton ky person

Numer_Telefoni – Numri i telefonit

Entiteti: Pasagjer, përfaqëson një pasagjer të thjeshtë të metrosë.

Atributet:

Entiteti: Punonjës, përfaqëson një punonjës të një stacioni.

Atributet:

Pozicion – pozicioni i punës së një personi

Orët_E_Punës – orët që një punonjës person bën çdo ditë.

Data_E_Punësimit – Data e punësimit të një punonjësi.

SUBKLASAT:

Shofer – Certifikata

Konduktor – Niveli_GCSE

Inxhinier – Niveli

Menaxher – Niveli_GCSE

Entiteti: Stacionet, ku ruhen të dhënat që përmban një stacion i metrosë.

Atributet:

ID – identifikuesi I një stacioni

Emër – emri I qytetit

Qyteti – Qyteti ku ndodhet stacioni

Rruga – Rruga që shërben si adresa e stacionit

Entiteti: Treni, ku ruhen të dhënat për cdo tren.

Atributet:

ID – Numri identifikues I një treni

Modeli – Modeli I një treni

Ngjyra – Ngjyra e trenit

Numër_Vendesh – Numri I vendeve në tren

Datë_Prodhimi – Data e prodhimit të trenit

Shpejtësi_Maksimale – Shpejtësia maksimale që mund të arrijë treni

Numër_Vagonash – Numri pjesëve ndarëse të trenit

Entiteti: Abonim, përfaqëson abonetë e blera nga persona të ndryshëm.

Atributet:

ID – numri identifikues i një aboneje

Cmimi – cmimi I abonesë

Numri_Udhëtimeve – Numri i udhëtimeve që mundëson aboneja

Klasa_E_Vendeve – Klasa ku do të udhëtojë pasagjeri

Entiteti: Bileta, përfaqëson informacionin në lidhje me një biletë

Atributet:

ID – numri identifikues i një bilete

Cmimi – cmimi I biletës

Data_E_Blerjes – Data e blerjes së biletës

Klasa – Klasa ku do të udhëtojë pasagjeri

Entiteti: Udhëtim, përfaqëson itinerarin e udhëtimeve

Atributet:

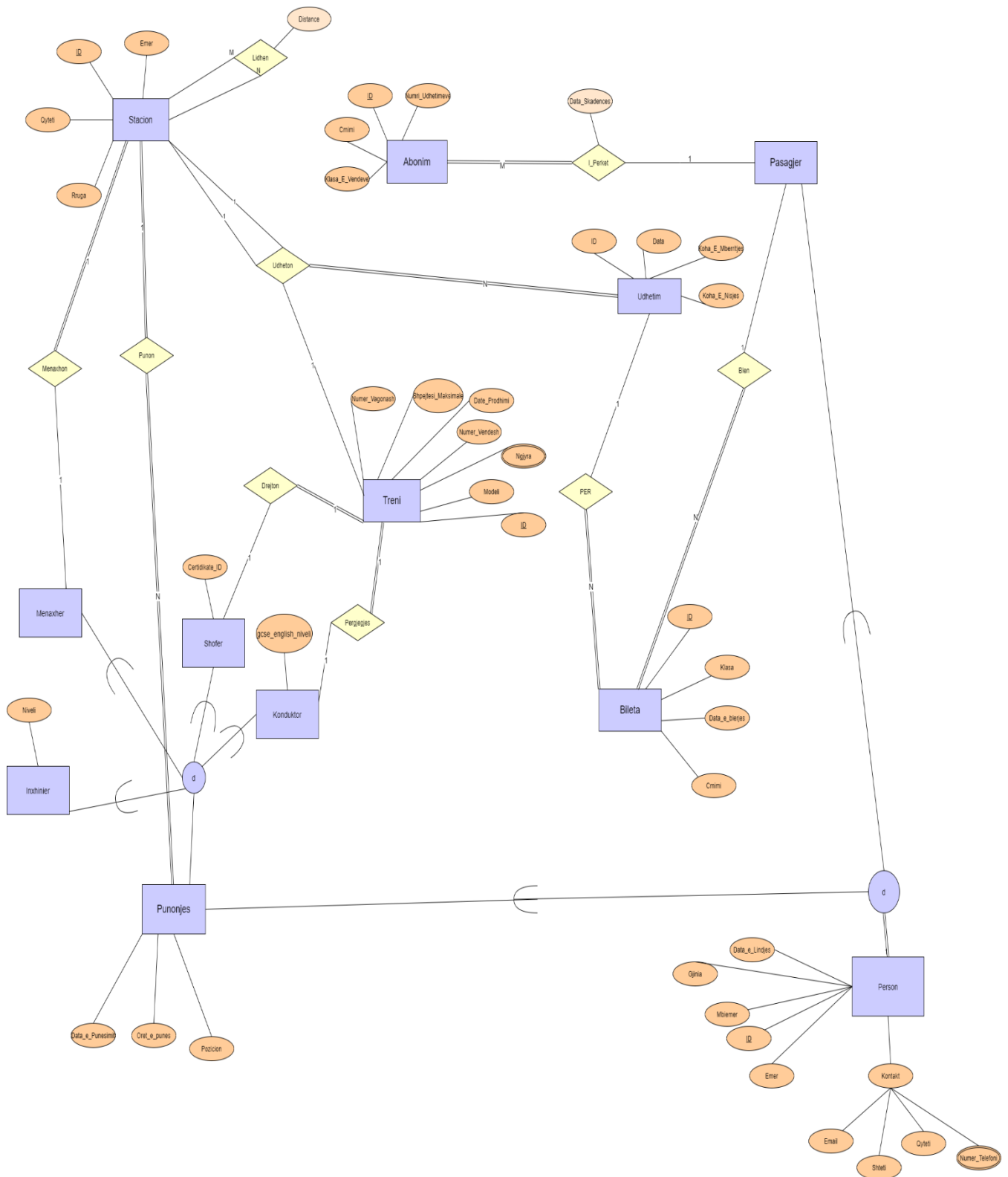
ID – numri identifikues i një bilete

Data – Data e udhëtimit

Koha_E_Nisjes

Koha_E_Mbërritjes

Skema ER/EER



Relacionet

Fillimisht do të shndërrojmë të gjithë entitetet e forta në tabela. Shohim se në rastin e entitetit **Person** ai ka dy subklasa **Punonjës** dhe **Pasagjer** për të cilat kufizimi I plotësisë është total. Pra në kalimin nga EER në relacional klasa **Person** nuk do të shndërrohet në relacion por vetëm subklasat e saj do të bëhen relacione duke trashëguar cdo atribut prej superklasës.

Skema jonë nuk përmban entitete të dobëta kështu që nuk duhet të mendojme për kalimin e celësve mes entiteteve të forta dhe të dobëta.

Atributet komplekse si kontakt zbërthehen në attribute atomike të cilat vendosen brenda relacioneve. Atributet shumëvlerëshe shndërrohen në relacione më vete dhe mbajnë ID e relacionit për të cilin ato ruajnë të dhënë specifike.

Nisim me marrdhëniet e tipit 1:1. Këto marrdhënie janë **Menaxhon** mes entitetit **Menaxher** dhe **Stacion** ku cdo për një stacion kemi vetëm një menaxher dhe për një menaxher kemi vetëm një stacion të cilin ai e menaxhon, **Drejton**, **Përgjegjës** mes **Shofer**, **Konduktor** dhe **Tren** respektivisht ku për cdo **Tren** kemi vetëm një **Shofer** dhe për cdo **Shofer** vetëm një tren që ai drejton si dhe për cdo **Konduktor** kemi vetëm një tren për të cilin ai është përgjegjës dhe për cdo **Tren** kemi një konduktor që e menaxhon atë.

Marrdhëniet e tipit 1:N janë :

Punon mes **Stacion** dhe **Punonjës** ku për një stacion mund të kemi disa punonjës(N) ndërkohe për një punonjës kemi vetëm një stacion ku ai punon(1).

I_Perket mes **Abonim** dhe **Pasagjer** ku për një abonim kemi një pasagjer(N) ndërsa për një pasagjer mund të kemi disa abonime në emrin e tij(N).

Blen mes **Pasagjer** dhe **Biletë** ku cdo pasagjer mund të blejë shumë bileta(N) ndërsa një biletë mund ti përkasë vetëm një pasagjeri(1).

Per mes **Udhëtim** dhe **Biletë** ku për cdo udhëtim kemi shumë bileta(N) ndërsa një biletë shërben për të realizuar vetëm një udhëtim.

Për të gjithë marrdhëniet e mësipërme që te shmangim teprinë e informacionit kalojmë celësin primar të relacionit(1) dhe atributet e marrdhënies brenda relacionit(N).

Marrdhënia **Udhëton** është e tipit 1:1:1:N sepse për një udhëtim, një stacion dhe një tren cfarëdo në një orar të caktuar mund të kemi vetëm një stacion(nisje/mbërritje). Për një stacion nisje, një stacion mbërritje dhe një udhëtim të caktuar mund të kemi vetëm një tren që e realizon këtë udhëtim. Ndërsa për dy stacione specifike, një tren të caktuar mund të kemi shumë udhëtime në orare të ndryshme. Për të shmangur krijimin e një relacioni për marrdhënien kalojmë celësat primarë të stacioneve dhe trenit brenda relacionit udhëtim.

Së fundmi kemi marrdhënien **Lidhen** mes dy stacioneve që është e tipit shumë me shumë pasi cdo stacion mund të lidhet me shumë stacione të tjere. Për këtë marrdhënie rekursive krijojmë një relacion më vete pasi nuk mund të kalojmë celësat primarë të asnjërit prej relacioneve tek tjetri pasi asl nuk e identifikon unikisht këtë marrdhënie.

Përsa i përket klasave **Menaxher, Inxhinier, Shofer** dhe **Konduktor** ato kalojnë secila në një relacion më vete ku mbajnë ID e punonjësit të cilin ato referojnë dhe atributet e tyre specifike.

Bazuar në relacionet e mësipërme ndërtohet dhe modeli relacional si më poshtë :

ENTITETE -> Relacion

1. Stacion(ID, Emer, Qyteti, Rruga)
2. Abonim(ID, Cmimi, Klasa_E_Vendeve, Numri_I_Udhetimeve)
3. Pasagjer(ID, Emer, Mbiemer, Shteti, Qyteti, Email)
4. Pasagjer_Kontakt(ID_Pasagjeri, Numer_Telfoni)
5. Punonjes(ID, Emer, Mbiemer, Data_E_Lindjes, Data_E_Punesimit, Oret_E_Punes, Pozicioni, Gjinia, Email, Qyteti, Shteti, Stacion_ID)
6. Punonjes_Kontakt(ID, Numer_Telefoni)
7. Udhetim(ID, Data, Koha_E_Nisjes, Koha_E_Mberritjes, Stacion_Nisje_ID, Stacion_Destinacion_ID, Treni_ID)
8. Bilete(ID, Klasa, Data_E_Blerjes, Cmimi, Pasagjer_ID, Udhetim_ID)
9. Treni(ID, Modeli, Statusi, Numer_Vendesh, Date_Prodhimi, Shpejtesia_Maksimale, Numer_Vagonash, Shofer_ID, Konduktor_ID)
10. Ngjyra_E_Trenit(ID, Ngjyra)
11. Menaxher(ID, Stacion_ID)
12. Inxhinier(ID)
13. Shofer(ID, certifikate_id)
14. Konduktor(ID, gcse_anglisht_niveli)

RELATIONSHIPS -> Relacion

15. [Lidhen] Rruge(Stacion_1ID, Stacion_2ID, Distance)
16. [Abonohet] Abonim_Pasagjeri(Abonim_ID, Pasagjer_ID, Data_Skadences)

Krijimi I Bazës Së Të Dhënave

Bazuar në diagramën ER/EER dhe skemën relacionale, bëjmë shtimin e të dhënave në databazë:

```
1  CREATE TABLE Pasagjer(  
2      p_id    INTEGER GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1,  
3      emer    VARCHAR(10),  
4      mbiemer VARCHAR(10),  
5      shtet   VARCHAR(10) NOT NULL,  
6      qytet   VARCHAR(15) NOT NULL,  
7      email   VARCHAR(30),  
8      CONSTRAINT pas_email UNIQUE(email),  
9      PRIMARY KEY(p_id)  
10 );  
11  
12  
13 CREATE TABLE Pasagjer_kontakt(  
14     p_id INTEGER,  
15     nr   VARCHAR(15) NOT NULL,  
16     PRIMARY KEY(p_id, nr),  
17     FOREIGN KEY(p_id) REFERENCES Pasagjer(p_id)  
18 );  
19  
20 CREATE TABLE Stacion  
21 (  
22     stacion_id NUMBER(6),  
23     emer       VARCHAR(30),  
24     qyteti     VARCHAR(20)  
25     CONSTRAINT stc_qytet_nn NOT NULL,  
26     rruga      VARCHAR(25)  
27     CONSTRAINT stc_rruge_nn NOT NULL,  
28     CONSTRAINT stc_id_pk PRIMARY KEY (stacion_id),  
29     CONSTRAINT emer_uk UNIQUE (emer)  
30 );
```

```
69 CREATE TABLE Rruge  
70 (  
71     stacion_1id NUMBER(6),  
72     stacion_2id NUMBER(6) ,  
73     distance NUMBER(4,2) CHECK (distance > 0),  
74     CONSTRAINT fk1_rruge FOREIGN KEY (stacion_1id) REFERENCES Stacion(stacion_id),  
75     CONSTRAINT fk2_rruge FOREIGN KEY (stacion_2id)REFERENCES Stacion(stacion_id),  
76     CONSTRAINT rruge_pk PRIMARY KEY(stacion_1id,stacion_2id)  
77 );  
78  
79 INSERT ALL
```



```

91 CREATE TABLE Punonjes (
92     punonjes_id NUMBER(6),
93     emer VARCHAR(20) NOT NULL,
94     mbiemer VARCHAR(20) NOT NULL,
95     data_e_lindjes DATE,
96     data_e_punesimit DATE NOT NULL,
97     oret_e_punes INTEGER,
98     pozicioni VARCHAR(50) NOT NULL,
99     gjinia VARCHAR(10),
100    email VARCHAR(50) NOT NULL,
101    qyteti VARCHAR(20) NOT NULL,
102    shteti VARCHAR(20) NOT NULL,
103    stacion_id NUMBER(6),
104    CONSTRAINT Punonjes_PK PRIMARY KEY (punonjes_id),
105    CONSTRAINT Stacion_ID_FK FOREIGN KEY (stacion_id) REFERENCES Stacion (stacion_id) ON DELETE SET NULL
106 );
107

```

```

211 CREATE TABLE Punonjes_Kontakt (
212     punonjes_id NUMBER(5) NOT NULL,
213     numer_kontakt VARCHAR(50) NOT NULL,
214     CONSTRAINT Punonjes_Kontakt_PK PRIMARY KEY (punonjes_id, numer_kontakt),
215     CONSTRAINT Punonjes_Kontakt_FK FOREIGN KEY (punonjes_id) REFERENCES Punonjes(punonjes_id) ON DELETE CASCADE
216 );
217

```

```

318 CREATE TABLE Menaxher
319 (
320     stacion_mng NUMBER(6),
321     m_id NUMBER(6),
322     CONSTRAINT mngr_stc_fk FOREIGN KEY(stacion_mng) REFERENCES Stacion(stacion_id),
323     CONSTRAINT mngr_id_fk FOREIGN KEY(m_id) REFERENCES Punonjes(punonjes_id),
324     CONSTRAINT mngr_pk PRIMARY KEY(m_id, stacion_mng)
325 );
326

```

```

349 CREATE TABLE Shofer(
350     ID Number(5) PRIMARY KEY,
351     Certifikate_ID VARCHAR(4),
352     CONSTRAINT shofer_pk FOREIGN KEY(ID) REFERENCES Punonjes(punonjes_id),
353     CONSTRAINT cert_valid_id CHECK(REGEXP_LIKE(Certifikate_ID, '^A\d{3}$'))
354 );
355

```

```

366 CREATE TABLE Konduktor(
367     ID Number(5) PRIMARY KEY,
368     GCSE_niveli_anglisht INTEGER,
369     CONSTRAINT konduktor_pk FOREIGN KEY(ID) REFERENCES Punonjes(punonjes_id),
370     CONSTRAINT niveli_valid CHECK(GCSE_niveli_anglisht >= 1 AND GCSE_niveli_anglisht <=5)
371 );
372

```

```

383 CREATE TABLE Inxhinier (
384     inxhinier_id NUMBER(6),
385     niveli VARCHAR(6),
386     CONSTRAINT Inxhinier_PK PRIMARY KEY (inxhinier_id, niveli),
387     CONSTRAINT Inxhinier_FK FOREIGN KEY (inxhinier_id) REFERENCES Punonjes(punonjes_id) ON DELETE CASCADE
388 );
389

```

```

416 CREATE TABLE Tren (
417     ID NUMBER(5) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1,
418     Model VARCHAR(50) NOT NULL,
419     Status SMALLINT,
420     Numer_Vendesh INTEGER NOT NULL,
421     Date_Prodhimi DATE DEFAULT(sysdate),
422     Shpejteshi_Maksimale FLOAT(1),
423     Numer_Vagonash INTEGER,
424     Shofer_ID NUMBER(5),
425     Konduktor_ID number(5),
426     CONSTRAINT shofer_id_fk FOREIGN KEY (Shofer_ID) REFERENCES Shofer(ID) ON DELETE SET NULL,
427     CONSTRAINT konduktor_id_fk FOREIGN KEY (Konduktor_ID) REFERENCES Konduktor(ID) ON DELETE SET NULL,
428     CONSTRAINT tren_pk PRIMARY KEY(ID),
429     CHECK(Shpejteshi_Maksimale > 0 AND Shpejteshi_Maksimale < 500)
430 );
431

```

```

463 CREATE TABLE Ngjyra_E_Trenit (
464     ID NUMBER(5) NOT NULL,
465     Ngjyra VARCHAR(10) NOT NULL,
466     CONSTRAINT Ngjyra_Trenit_PK PRIMARY KEY (ID,Ngjyra),
467     CONSTRAINT Tren_ID_FK FOREIGN KEY (ID) REFERENCES Tren(ID) ON DELETE CASCADE
468 );
469

```

```

487 CREATE TABLE Udhetim (
488     ID NUMBER(5) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1,
489     Koha_E_Nisjes TIMESTAMP NOT NULL,
490     Koha_E_Mberritjes TIMESTAMP NOT NULL,
491     Stacion_Nisje_ID NUMBER(5) NOT NULL,
492     Stacion_Destacion_ID Number(5) NOT NULL,
493     Tren_ID Number(5) NOT NULL,
494     CONSTRAINT udhetim_pk PRIMARY KEY (ID),
495     CONSTRAINT stacion_nisje_fk FOREIGN KEY (Stacion_Nisje_ID) REFERENCES Stacion(stacion_id) ON DELETE CASCADE,
496     CONSTRAINT stacion_mberritje_fk FOREIGN KEY (Stacion_Destacion_ID) REFERENCES Stacion(stacion_id) ON DELETE CASCADE,
497     CONSTRAINT tren_fk FOREIGN KEY (Tren_ID) REFERENCES Tren(ID) ON DELETE CASCADE
498 );
499

```

```

625 CREATE TABLE Abonim (
626     abonim_id NUMBER(5) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1,
627     cmimi FLOAT(2) NOT NULL,
628     klasa_e_vendeve VARCHAR(10),
629     numri_i_udhetimeve INTEGER,
630     CONSTRAINT Abonim_PK PRIMARY KEY(abonim_id)
631 );
632

```

```

638 CREATE TABLE Abonim_Pasagjeri(
639     abonim_id INTEGER,
640     p_id INTEGER,
641     dts DATE NOT NULL,
642     PRIMARY KEY(abonim_id, p_id),
643     FOREIGN KEY(abonim_id) REFERENCES Abonim(abonim_id),
644     FOREIGN KEY(p_id) REFERENCES Pasagjer(p_id)
645 );
646

```

```

652 CREATE TABLE Bilete
653 (
654     bilete_id NUMBER(7) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1,
655     klasa     VARCHAR(10),
656     data_blerjes TIMESTAMP NOT NULL,
657     cmimi     NUMBER(4,2) CONSTRAINT bilete_cm_ck CHECK (cmimi > 0),
658     CONSTRAINT bilete_id_pk PRIMARY KEY (bilete_id),
659     pasagjer_id NUMBER(5),
660     udhetim_id NUMBER(9),
661     CONSTRAINT bilete_psg_id_fk FOREIGN KEY(pasagjer_id) REFERENCES Pasagjer(p_id),
662     CONSTRAINT bilete_udht_id_fk FOREIGN KEY(udhetim_id) REFERENCES Udhetim(ID)
663 );
664

```

Query

ALBI TAULLA

HISTOGRAMME PER CMIMET E BILETAVE

```
SELECT cmimi,COUNT(*)  
FROM Bilete  
GROUP BY cmimi  
ORDER BY 1;
```

PUNONJESIT QE KANE DALE NE PENSION OSE JANE 6 MUAJ PARA DALJES NE PENSION

```
SELECT punonjes_id,emer,mbiemer,age  
FROM(  
    SELECT punonjes_id,emer,mbiemer,gjinia,data_e_lindjes,  
    CASE WHEN EXTRACT(month FROM data_e_lindjes) < EXTRACT(month FROM SYS  
        DATE) OR  
    (EXTRACT(month FROM data_e_lindjes) = EXTRACT(month FROM SYSDATE)  
    AND (EXTRACT(day FROM data_e_lindjes) <= EXTRACT(day FROM SYSDATE)))  
    THEN EXTRACT(year from sysdate) - EXTRACT(year FROM data_e_lindjes)  
    ELSE EXTRACT(year from sysdate) - EXTRACT(year FROM data_e_lindjes) - 1 END AS  
        age  
    FROM Punonjes  
)  
WHERE (gjinia = 'M' AND (AGE >= 64 OR (AGE = 63 AND EXTRACT(month FROM  
data_e_lindjes) - EXTRACT(month FROM SYSDATE) <=6 )))  
OR (gjinia = 'F' AND (AGE >= 62 OR (AGE = 61 AND EXTRACT(month FROM  
data_e_lindjes) - EXTRACT(month FROM SYSDATE) <=6 )));
```

LISTA E PUNONJESVE NE CDO STACION

```
SELECT P.punonjes_id,P.emer,p.mbiemer,P.pozicioni,S.emer  
FROM Punonjes P INNER JOIN Stacion S ON P.stacion_id = S.stacion_id  
WHERE S.stacion_id = 1;
```

UPDATIMI I SHOFRERIT APO KONDUKTORIT TE NJE TRENI

```
UPDATE Tren
SET Shofer_id = 17
WHERE ID = 9;

UPDATE Tren
SET Konduktor_id = 6
WHERE ID = 10;
```

UPDATIMI I KOHES SE NISJES DHE MBERRITJES PER NJE UDHETIM

```
UPDATE Udhetim
SET Koha_E_Nisjes = X, Koha_E_Mberritjes = Y
WHERE ID = 5;
```

FSHIRJA E TRENAVE ME STATUS PASIV APO NJE TRENI ME ID TE CAKTUAR

```
DELETE FROM Tren
WHERE Status = 0;

DELETE FROM Tren
WHERE ID = 1;
```

FSHIRJA E NJE UDHETIMI ME ID TE DHENE

```
DELETE FROM Udhetim
WHERE ID = 1;
```

MUAJI ME FREKUENCEN ME TE LARTE TE UDHETIMEVE NE VITIN 2014

```
SELECT month, COUNT(*) AS frequency
FROM (
    SELECT TO_CHAR(CAST(Koha_E_Nisjes AS DATE), 'MONTH') AS month
    FROM Udhetim
    WHERE EXTRACT(year FROM CAST(Koha_E_Nisjes AS DATE)) = 2014
)
```

```
GROUP BY month
ORDER BY frequency DESC
FETCH FIRST 1 ROWS ONLY;
```

UDHETIMET MES DY STACIONEVE QE NUK KANE KALUAR ENDE

```
SELECT ID,TO_CHAR(Koha_E_Nisjes, 'MONTH DD, YYYY "at" HH24:MI') Nisja,
TO_CHAR(Koha_E_Mberritjes, 'MONTH DD, YYYY "at" HH24:MI') Mberritja
FROM Udhetim
WHERE Stacion_Nisje_ID = 1 AND Stacion_Destacion_ID =4 AND
EXTRACT(year FROM Koha_E_Nisjes) = EXTRACT(year FROM SYSDATE) AND
EXTRACT(month FROM Koha_E_Nisjes) >= EXTRACT(month FROM SYSDATE) AND
EXTRACT(day FROM Koha_E_Nisjes) >= EXTRACT(day FROM SYSDATE);
```

TE GJITHA BILETAT QE KA BLERE PASAGJERI ME NUMER X(3 NE RASTIN TONE)

```
SELECT P.emer, B.bilete_id,B.klasa,TO_CHAR(B.Data_Blerjes, 'MONTH DD, YYYY "AT" HH
24:MI'),B.cmimi
FROM Pasagjer P INNER JOIN Bilete B ON P.p_id = B.pasagjer_id
WHERE P.p_id = 3;
```

KONDUKTORI I CDO TRENI

```
SELECT K.ID ID_Konduktori, P.emer,P.mbiemer, T.ID ID_Treni, T.Model
FROM (Konduktor K INNER JOIN Punonjes P ON K.ID = P.punonjes_id) INNER JOIN Tren T O
N T.Konduktor_ID = K.ID;
```

IRIS DIZDARI

TE GJITHA ABONIMET E NJE PASAGJERI X

```
SELECT abonim_id, P.p_id, emer, mbiemer
FROM Abonim_Pasagjeri A, Pasagjer P
WHERE A.p_id = 1 AND A.p_id = P.p_id
```

MENAXHERI I NJE STACIONI

```
SELECT P.punonjes_id, P.emer, P.mbiemer
FROM Punonjes P
WHERE P.punonjes_id IN(
    SELECT M.m_id
    FROM Menaxher M
    WHERE stacion_mng IN (
        SELECT S.stacion_id
        FROM Stacion S
        WHERE S.emer = 'Glendale/19th Avenue'
    )
)
```

STACIONET E NJE TRENI

```
SELECT U.ID, U.Stacion_Nisje_ID, S.emer
FROM Udhetim U, Stacion S
WHERE U.Stacion_Nisje_D = S.stacion_id AND U.Treni_id = 6;
```

DITA QE JANE SHITUR ME SHUME BILETA

```
SELECT data_blerjes, COUNT(data_blerjes)
FROM Bilete GROUP BY data_blerjes
HAVING COUNT (data_blerjes)=(
    SELECT MAX(COUNT)
    FROM (
        SELECT data_blerjes, COUNT(data_blerjes) COUNT
        FROM Bilete
        GROUP BY data_blerjes))
```

ANULLIMI I ABONIMIT

```
DELETE Abonim_Pasagjeri A
WHERE A.dts = '14-MAR-2018';
```

INSERT RRUGE MES DY STACIONEVE

```
INSERT INTO Rruge VALUES(007, 009, 2.95);
```

UPDATE PASAGJER

```
UPDATE Pasagjer  
SET qytet = 'Florence'  
WHERE p_id = 1
```

ALBI NDROQI

PUNONJESI QE KA PUNUAR ME SHUME ORE PER NJE VIT

```
SELECT emer, mbiemer, oret_e_punes*5*50 "Ore pune ne vit"  
FROM Punonjes  
WHERE oret_e_punes = (SELECT MAX(oret_e_punes)  
FROM Punonjes);
```

STACIONET QE LIDHEN DREJPERDREJT

```
SELECT s.emer , s2.emer , distance || ' km' AS "Distanca"  
FROM Stacion s, Rruge, Stacion s2  
WHERE (s.stacion_id = rruge.stacion_1id)  
AND (s2.stacion_id = rruge.stacion_2id);
```

NUMRI I KONTAKTIT TE MENAXHERIT PER NJE STACION X

```
SELECT numer_kontakt  
FROM Punonjes_Kontakt  
WHERE punonjes_id IN (SELECT m_id  
FROM Menaxher
```



```
WHERE stacion_mng = &id_STACIONIT);
```

PASAGJERI QE E KALON NUMRIN E UDHETIMEVE

```
SELECT p.emer, p.mbiemer, ab.numri_i_udhetimeve
FROM Pasagjer p , Abonim_Pasagjeri a, Abonim ab
WHERE (p.p_id = a.p_id)
AND (ab.abonim_id = a.abonim_id
AND ab.numri_i_udhetimeve < (SELECT COUNT(*)
FROM Bilete b
WHERE b.pasagjer_id = p.p_id));
```

LISTA E PASAGJEREVE QE I SKADON ABONIMI PER 1, 3, 6 MUAJ

```
SELECT *
FROM Abonim_Pasagjeri
WHERE MONTHS_BETWEEN(dts,SYSDATE) = 1;
```

```
SELECT *
FROM Abonim_Pasagjeri
WHERE MONTHS_BETWEEN(dts,SYSDATE) = 3;
```

```
SELECT *
FROM Abonim_Pasagjeri
WHERE MONTHS_BETWEEN(dts,SYSDATE) = 6;
```

PASAGJERI QE KA UDHETUAR TE PAKTEN 3 HERE BREND A NJE JAVE TE CAKTUAR

```
SELECT p_id, emer, mbiemer
FROM Pasagjer
GROUP BY p_id, emer, mbiemer
HAVING ((SELECT COUNT(*)
FROM Bilete
WHERE p_id = pasagjer_id
AND (Data_Blerjes BETWEEN '11-AUG-20' AND '18-AUG-20')) = 3;
```

FSHIRJE UDHETIMI

```
DELETE FROM Udhetim  
WHERE id = &ID_UDHETIMI;
```

LISTA E TRENAVE NE STATUS PASIV

```
SELECT ID, Model, Status  
FROM TREN  
WHERE (Status=0);
```

ERSA DOKA

NUMRI I BILETAVE NE NJE DITE PER NJE STACION

```
SELECT CAST(B.Data_Blerjes AS DATE) AS Date_Blerje, COUNT(*) as Sasia_e_Biletave  
FROM Bilete B  
INNER JOIN Udhetim U ON  
B.UDHETIM_ID = U.ID  
WHERE U.STACION_NISJE_ID = 4  
GROUP BY CAST(Data_Blerjes AS DATE)
```

UDHETIMET E TE GJITHE TRENAVE NE NJE ORAR X

```
SELECT ID ID_UDHETIMIT, KOHA_E_NISJES  
FROM UDHETIM  
WHERE CAST(KOHA_E_NISJES AS TIME) > '08.00.00 AM' AND CAST(KOHA_E_NISJES  
AS TIME) < '10.00.00 AM'
```

PASAGJERI QE KA UDHETUAR ME SHUME NE NJE VIT X

```
SELECT PASAGJER_ID, COUNT(*) SASIA
FROM BILETE
GROUP BY PASAGJER_ID
HAVING COUNT(*) = (
    SELECT MAX(COUNT(*))
    FROM BILETE
    WHERE EXTRACT(YEAR FROM DATA_BLERJES) = '2020'
    GROUP BY PASAGJER_ID
)
```

SASIA E PUNONJESVE QE KA PUNUAR PART TIME DHE FULL TIME PER CDO STATION

```
SELECT STACION_ID, count(ORET_E_PUNES) as FULL_TIME
FROM PUNONJES
where ORET_E_PUNES = 4
GROUP BY STACION_ID
ORDER BY STACION_ID ASC
```

```
SELECT STACION_ID, count(ORET_E_PUNES) as FULL_TIME
FROM PUNONJES
where ORET_E_PUNES = 8
GROUP BY STACION_ID
ORDER BY STACION_ID ASC
```

UPDATE/FSHIRJE/INSERT NE ELEMENTET E TABELAVE

```
UPDATE PUNONJES
SET EMAIL = 'DUMMYMAIL@GMAIL.COM'
WHERE PUNONJES_ID = 5

UPDATE PUNONJES_KONTAKT
SET NUMER_KONTAKT = '06888888888'
WHERE PUNONJES_ID = 36
```

```
UPDATE ABONIM
SET CMIMI = 70
WHERE KLASA_E_VENDEVE = 'Student'

INSERT INTO ABONIM(ABONIM_ID, CMIMI, KLASA_E_VENDEVE,
NUMRI_I_UDHETIMEVE) VALUES (5, 10, 'OFERTE', 100)

DELETE ABONIM WHERE KLASA_E_VENDEVE = 'OFERTE'
```

LISTA E KONDUKTOREVE PER TOP 3 TRENAT ME TE SHPEJTE

```
SELECT ID "ID E TRENIT", SHPEJTESI_MAKSIMALE, KONDUKTOR_ID
FROM TREN
ORDER BY SHPEJTESI_MAKSIMALE DESC
FETCH FIRST 3 ROWS ONLY
```

IEWS

ALBI TAULLA

TE GJITHE UDHETIMET QE REALIZOHEN NE DITEN E SOTME

```
CREATE VIEW Udhetimet_E_Dites AS
SELECT S1.emer || ' Qyteti: ' || S1.qyteti || ' Rruga: ' || S1.rruga Stacioni1,
S2.emer || ' Qyteti: ' || S2.qyteti || ' Rruga: ' || S2.rruga AS Stacioni2,
U.ID Treni,TO_CHAR(U.Koha_E_Nisjes, 'MONTH DD, YYYY "at" HH24:MI') Nisja,
TO_CHAR(U.Koha_E_Mberritjes, 'MONTH DD, YYYY "at" HH24:MI') Mberritja
FROM Udhetim U
INNER JOIN Stacion S1 ON U.Stacion_Nisje_ID = S1.stacion_id
INNER JOIN Stacion S2 ON U.Stacion_Destinacion_ID = S2.stacion_id
WHERE EXTRACT(year FROM U.Koha_E_Nisjes) = EXTRACT(year FROM SYSDATE)
AND
EXTRACT(month FROM U.Koha_E_Nisjes) >= EXTRACT(month FROM SYSDATE) AND
EXTRACT(day FROM U.Koha_E_Nisjes) >= EXTRACT(day FROM SYSDATE);
```

NUMRI I UDHETIMEVE QE KA KRYER CDO TREN

```
CREATE VIEW Udhetimet_Per_Tren AS
SELECT T.ID,T.Model,COUNT(*) frequency
FROM Tren T INNER JOIN Udhetim U ON T.ID = U.Tren_ID
GROUP BY T.ID,T.Model
ORDER BY frequency DESC;
```

IRIS DIZDARI

MESATARJA E PASAGJEREVE QE UDHETOJNE ME TREN PER X MUAJ

```
CREATE VIEW Bilete_Pasagjer AS

SELECT AVG(Bilete.bilete_id)

FROM Pasagjer, Bilete

WHERE Bilete.data_blerjes < '01-JUL-2020'
```

TE GJITHE PASAGJERET QE NUK KANE UDHETUAR ASNJEHERE ME NJE TREN X

```
CREATE VIEW Pasagjer_Tren AS
SELECT P.p_id, P.emer, P.mbiemer
FROM Pasagjer P
WHERE P.p_id NOT IN(
    SELECT DISTINCT B.pasagjer_id
    FROM Bilete B, Udhetim U
    WHERE B.udhetim_id = U.ID AND U.Tren_ID = 9
)
```

ALBI NDROQI

MENAXHERET E STACIONIT 1

```
CREATE VIEW Menaxher_Stacion_1 AS
SELECT p.emer, p.mbiemer
FROM Punonjes p
WHERE p.punonjes_id IN(SELECT punonjes_id
    FROM Punonjes_Kontakt
    WHERE punonjes_id IN (SELECT m_id
    FROM Menaxher
    WHERE stacion_mng = 1));
```

STACIONET ME DISTANCEN MAKSIMALE

```
CREATE VIEW Stacion_Distance_MAX AS
SELECT s1.emer AS "Emri Stacionit 1", s2.emer AS "Emri Stacionit 2"
FROM Stacion s1, Stacion s2
WHERE s1.stacion_id = (SELECT stacion_1id
    FROM Ruge
    WHERE distance = (SELECT MAX(distance)
    FROM Ruge))
AND s2.stacion_id = (SELECT stacion_2id
    FROM Ruge
    WHERE distance = (SELECT MAX(distance)
    FROM Ruge));
```

ERSA DOKA

BILETATE CDO PASAGJERI

```
CREATE VIEW BILETAT_E_CDO_PASAGJERI AS
  SELECT PASAGJER_ID, COUNT(*) SASIA_E_BILETAVE
  FROM BILETE
  GROUP BY PASAGJER_ID
  ORDER BY SASIA_E_BILETAVE DESC
```

I PUNESUARI I PARË

```
CREATE VIEW I_PUNESUARI_I_PARE AS
  SELECT *
  FROM PUNONJES
  WHERE DATA_E_PUNESIMIT = (
    SELECT MIN(DATA_E_PUNESIMIT)
    FROM PUNONJES
  )
```

INDEXET

ALBI TAULLA

```
CREATE INDEX Stacion_ID_Index ON Stacion(stacion_id);  
CREATE INDEX Tren_ID_Index ON Tren(ID);
```

ALBI NDROQI

```
CREATE INDEX BILETE_DB_INDEX ON Bilete(Data_Blerjes);
```

ERSA DOKA

```
CREATE INDEX PUNONJES_ID_INDEX ON PUNONJES(PUNONJES_ID)
```