

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΤΙΜΩΝ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

ΕΡΓΑΣΙΑ 2

ΟΜΑΔΑ 1

Κύρκα Μαρίνα-Βασιλική	58546
Μπουρνουσούζης Παρασκευάς	58746
Φραγκόπουλος Αναστάσιος	58633

Ερώτημα 1.1:

Στο σύστημα υπάρχουν καταναλωτές και πάροχοι ηλεκτρικής ενέργειας.

Ο κάθε πάροχος ηλεκτρικής ενέργειας έχει όνομα, τηλέφωνο και email. Ο κάθε καταναλωτής έχει όνομα, επώνυμο, email, κινητό τηλέφωνο, σταθερό τηλέφωνο και δεδομένα πιστωτικής κάρτας.

Οι πάροχοι διαθέτουν διάφορα προγράμματα και το καθένα από αυτά χαρακτηρίζεται από ένα όνομα, την τιμή που αντιστοιχεί σε μία κιλοβατώρα, τον μήνα και το έτος έκδοσής του, το είδος του (σταθερό, κυμαινόμενο, ειδικό) και τη διάρκειά του.

Σε κάθε καταναλωτή ανήκουν ένας ή περισσότεροι μετρητές, στους οποίους αναγράφονται οι κιλοβατώρες που καταναλώθηκαν και ο μοναδικός αριθμός παροχής. Επίσης, ο κάθε μετρητής εμπεριέχει ένδειξη σύνδεσης με το δίκτυο και χαρακτηρίζεται από την ονομαστική του ισχύ και τη διεύθυνση στην οποία βρίσκεται. Οι καταναλωτές επιλέγουν ένα πρόγραμμα για κάθε έναν από τους μετρητές τους.

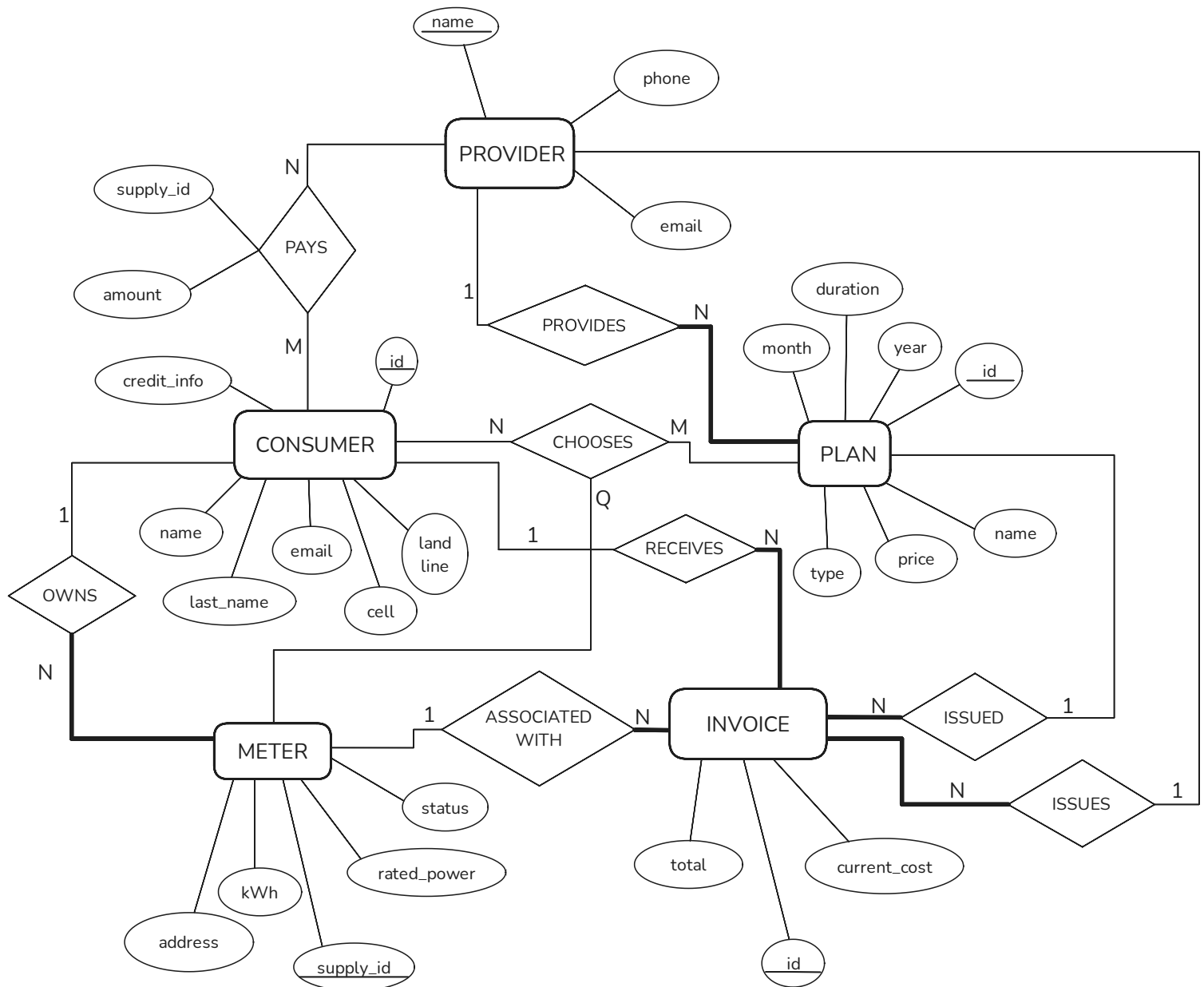
Κάθε μήνα, οι πάροχοι εκδίδουν τιμολόγια στους καταναλωτές για τους μετρητές τους ανάλογα με το πλάνο που έχει επιλεγεί για τον κάθε μετρητή. Στα τιμολόγια φαίνεται ο αριθμός έκδοσής τους, το οφειλόμενο ποσό που αντιστοιχεί στην κατανάλωση του μήνα έκδοσης του τιμολογίου και το συνολικό οφειλόμενο ποσό.

Οι καταναλωτές μπορούν να πληρώσουν τους παρόχους οποιοδήποτε ποσό σε οποιαδήποτε στιγμή και για οποιονδήποτε μετρητή.

Γίνεται η υπόθεση ότι με κάθε έκδοση τιμολογίου για κάποιον μετρητή, μηδενίζεται η ένδειξη κατανάλωσής του.

Ερώτημα 1.2:

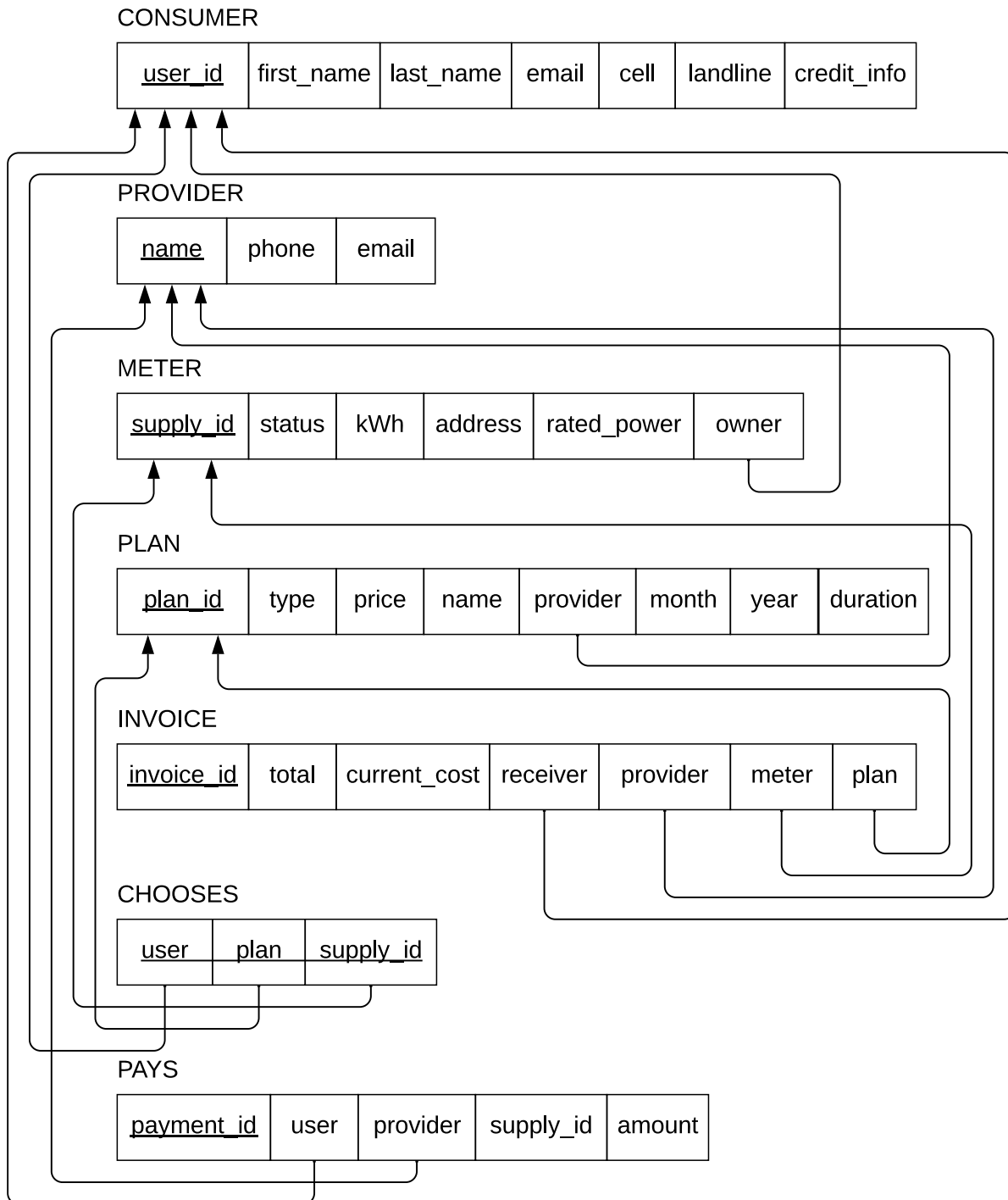
Παρακάτω παρουσιάζεται το διάγραμμα οντοτήτων-συσχετίσεων που αντιστοιχεί στις απαιτήσεις της βάσης δεδομένων που παρουσιάστηκαν στο προηγούμενο ερώτημα:



Διάγραμμα 1: Διάγραμμα Οντοτήτων-Συσχετίσεων

Ερώτημα 2.1:

Παρακάτω παρουσιάζεται το σχεσιακό σχήμα που προέκυψε από το διάγραμμα οντοτήτων-συσχετίσεων:



Διάγραμμα 2: Σχεσιακό Σχήμα της Βάσης Δεδομένων

Ερώτημα 2.2:

Ο κώδικας για τη δημιουργία του σχήματος της βάσης δεδομένων, καθώς και τα αρχεία που χρησιμοποιήθηκαν για την εισχώρηση δεδομένων στη βάση βρίσκονται στο [GitHub](#). Το σχήμα της βάσης δημιουργήθηκε με τις παρακάτω εντολές:

```
CREATE DATABASE lab2425omada1_EPMS;
USE lab2425omada1_EPMS;
ALTER DATABASE lab2425omada1_EPMS
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;

CREATE TABLE CONSUMER (
    user_id int not null AUTO_INCREMENT,
    first_name varchar(50) not null,
    last_name varchar(50) not null,
    email varchar(62) not null UNIQUE,
    cell varchar(10) not null UNIQUE,
    landline varchar(10) UNIQUE,
    credit_info int,

    primary key (user_id)
);

CREATE TABLE PROVIDER (
    name varchar(50) not null,
    phone varchar(10) not null UNIQUE,
    email varchar(50) not null UNIQUE,

    primary key (name)
);

CREATE TABLE METER (
    supply_id int not null AUTO_INCREMENT,
    status bool DEFAULT 0,
    kWh int DEFAULT 0,
    address varchar(100) not null,
    rated_power int not null,
    owner int not null,

    primary key (supply_id),
    foreign key (owner) references CONSUMER(user_id)
);
```

```

CREATE TABLE PLAN (
    plan_id int not null AUTO_INCREMENT,
    type varchar(40) not null,
    price float not null,
    name varchar(50),
    provider varchar(50) not null,
    month varchar(30) not null,
    year int not null,
    duration int not null,

    primary key (plan_id),
    foreign key (provider) references PROVIDER(name)
);

/*
 * id int unsigned: bigger number
 * auto_increament: use LAST_INSERT_ID()
 * ALTER TABLE tbl AUTO_INCREMENT = num; to start from bigger number
 */
CREATE TABLE INVOICE (
    invoice_id int not null AUTO_INCREMENT,
    total float DEFAULT 0,
    current_cost float not null,
    receiver int not null,
    meter int not null,
    provider varchar(50) not null,
    plan int not null,

    primary key (invoice_id),
    foreign key (receiver) references CONSUMER(user_id),
    foreign key (provider) references PROVIDER(name),
    foreign key (meter) references METER(supply_id),
    foreign key (plan) references PLAN(plan_id)
);

CREATE TABLE CHOOSES (
    user int not null,
    plan int not null,
    supply_id int not null,

    primary key (user, plan, supply_id),
    foreign key (user) references CONSUMER(user_id),
    foreign key (plan) references PLAN(plan_id),
    foreign key (supply_id) references METER(supply_id)
);

```

```
CREATE TABLE PAYS (
    payment_id int not null AUTO_INCREMENT,
    user int not null,
    provider varchar(50) not null,
    supply_id int not null,
    amount float DEFAULT 0,

    primary key (payment_id),

    foreign key (user) references CONSUMER(user_id),
    foreign key (provider) references PROVIDER(name)
);
```

Για την εισχώρηση δεδομένων στη βάση, χρησιμοποιήθηκε το αρχείο κώδικα *app.py*, το οποίο δημιουργεί εντολές εισχώρησης δεδομένων. Παρακάτω, δίνονται οι παραμετροποιημένες εντολές, όπως αυτές μπορούν να εντοπιστούν στο αρχείο κώδικα:

► Εισχώρηση δεδομένων στον πίνακα CONSUMER:

```
curs.execute(f"""
INSERT INTO CONSUMER
(first_name, last_name, email, cell{landline[0]})
VALUES
("{cons.first_name}", "{cons.last_name}", "{cons.email}",
 "{cons.cell}" {landline[1]});
""")
```

► Εισχώρηση δεδομένων στον πίνακα METER:

```
curs.execute(f"""
INSERT INTO METER
(address, rated_power, owner)
VALUES
("{meter.address}", {meter.rated_power}, {meter.owner});
""")

curs.execute(f"""
UPDATE METER
SET status = {int(True)}
WHERE supply_id = {meter["supply_id"]};
""")
```

```

curs.execute(f"""
UPDATE METER
SET kWh = {kWh}
WHERE supply_id = {meter["supply_id"]};
""")

```

► Εισχώρηση δεδομένων στον πίνακα PROVIDER:

```

curs.execute(f"""
INSERT INTO PROVIDER
(name, phone, email)
VALUES
("{provider.name}", "{provider.phone}", "{provider.email}");
""")

```

► Εισχώρηση δεδομένων στον πίνακα PLAN:

```

curs.execute(f"""
INSERT INTO PLAN
(type, price, name, provider, month, year, duration)
VALUES
("{plan.type}", {plan.price}, "{plan.name}", "{plan.provider}",
"{months[plan.month-1].name}", {months[plan.month-1].year},
{plan.duration});
""")

```

► Εισχώρηση δεδομένων στον πίνακα CHOOSES:

```

curs.execute(f"""
INSERT INTO CHOOSES
(user, plan, supply_id)
VALUES
({meter["owner"]}, {random_plan["plan_id"]}, {meter["supply_id"]});
""")

curs.execute(f"""
UPDATE CHOOSES
SET plan = {random_plan["plan_id"]}
WHERE supply_id = {meter["supply_id"]};
""")

```


► Εισχώρηση δεδομένων στον πίνακα INVOICE:

```
curs.execute(f"""
INSERT INTO INVOICE
(total, current_cost, receiver, provider, meter, plan)
VALUES
({cost}, {cost}, {meter["owner"]}, "{random_plan["provider"]}",
{meter["supply_id"]}, {random_plan["plan_id"]});
""")
```

► Εισχώρηση δεδομένων στον πίνακα PAYS:

```
curs.execute(f"""
INSERT INTO PAYS
(user, provider, supply_id, amount)
VALUES
({meter["owner"]}, "{random_plan["provider"]}",
{meter["supply_id"]}, {cost});
""")
```

Ερώτημα 2.3:

Παρακάτω παρουσιάζονται έξι ενδεικτικά ερωτήματα προς τη βάση δεδομένων και οι αντίστοιχες περιπτώσεις στις οποίες μπορούν να χρησιμοποιηθούν. Επειδή τα δεδομένα στη βάση δεδομένων εμπεριέχουν και ελληνικούς χαρακτήρες, πρέπει μετά από τη σύνδεση στη βάση να σταλθεί η παρακάτω εντολή για να αλλάξει η κωδικοποίηση σε UTF-8, και τα δεδομένα να εμφανίζονται σωστά στον χρήστη.

```
SET character_set_client = "utf8mb4",
character_set_results = "utf8mb4",
character_set_connection = "utf8mb4";
```

► Να βρεθεί το πρόγραμμα που έχει επιλέξει ο καταναλωτής για έναν συγκεκριμένο αριθμό παροχής (π.χ. 57):

```
SELECT name, type, price, provider, month, year, duration
FROM CHOOSES, PLAN
WHERE plan = plan_id AND supply_id = 57;
```

- Να βρεθούν όλα τα διαθέσιμα προγράμματα για τον Νοέμβριο του 2024 με αύξουσα σειρά τιμών:

```
SELECT name, provider, price
FROM PLAN
WHERE year = 2024 AND month = "November" ORDER BY price;
```

- Να βρεθούν το όνομα και το τηλέφωνο του παρόχου, του οποίου το πρόγραμμα έχει επιλεγθεί για συγκεκριμένο αριθμό παροχής (π.χ. 57):

```
SELECT P.name, P.phone
FROM PLAN, CHOOSES, PROVIDER P
WHERE plan = plan_id AND provider = P.name AND supply_id = 57;
```

- Να βρεθούν όλα τα τιμολόγια που έχει εκδώσει ένας πάροχος (π.χ. ΔΕΗ) σε φθίνουσα σειρά οφειλόμενου ποσού:

```
SELECT invoice_id, total
FROM INVOICE
WHERE provider = "ΔΕΗ" ORDER BY total DESC;
```

- Να βρεθούν τα ονόματα και τα επίθετα όλων των καταναλωτών που έχουν πραγματοποιήσει πληρωμές προς τους παρόχους μεγαλύτερες από 150€ με αύξουσα σειρά ποσού πληρωμής:

```
SELECT first_name, last_name, amount
FROM PAYS, CONSUMER
WHERE user = user_id AND amount > 150 ORDER BY amount;
```

- Να βρεθούν τα ονόματα, τα επίθετα, οι αριθμοί παροχής και τα σταθερά τηλέφωνα όλων των καταναλωτών που έχουν σταθερό τηλέφωνο:

```
SELECT first_name, last_name, supply_id, landline
FROM CONSUMER, METER
WHERE owner = user_id AND landline IS NOT NULL;
```