Database Project

At Your Service

Website for the purchase/sale of products available through certain services by all categories of shopkeepers

edited by Stefano Di Patti

Requirements collection and analysis

Information and introduction to the project:

We want to create a platform through which customers can view and order products offered by all types of retailers through specific services made available, such as: home delivery, shipping, on-site collection, etc. ...

There are two types of users involved: **Activity** & **Customer**.

The business registers with the system by providing a VAT number, Password, Name, Address, Description, Working hours, telephone number, Link to the owner website, Logo and category (e.g. Clothing, Bar, etc.). The businesses make certain types of services available to proceed with the purchase of their products, such as: collection of goods on site, shipping, home delivery, rental, commitment. Furthermore, they establish which payment methods will be accepted among the various types available: payment on delivery, by card, PayPal, meal vouchers, virtual currency. The products sold are further divided into categories and are characterized by: code, price, description, availability and possible discount.

The purchase of products takes place through ordering by the customer. An order will therefore be characterized by: date, seller and customer, service and payment method selected by the customer from those available, total price and obviously the list of items ordered.

The customer accesses the system via email and password, but will have to enter their name, address and telephone to ensure availability if needed. As a feedback system, there is a system for customers to write reviews on the activity which will be made public on the page together with an attached rating on the quality of the purchase.

A business can be a customer of another company, so one can place orders by accessing exclusive discounts, since these are purchases of a VAT holder.

Terms' Glossary

Term	Description	Synonyms	Related terms
Activity Category	Grouping by type of activity	Kind of activity	Activity
Activity	Exercise of selling certain products	Activities, Shop, Shopping Center	Payment methods, Service, Products, Orders
Customer	Consumer purchasing products	Consumer, Purchaser, Buyer	Orders, Products, Payment Methods, Reviews
Service	Product delivery type	Performance, Delivery	Activities, Orders, Products, Customer
Payment mode	Tools through which the compensation is accepted by the activities	Salary, Remuneration	Order, Customer, Activity, Products, Product Category
Reviews	Customer reviews of the business	Judgment, Evaluation	Customer, Activities, Products, Orders, Service
Orders	Customer request for a business's products	Request, Commission	Customer, Activity, Payment Method, Service, Products
Product Category	Grouping by type of products	Type of supplies	Products
Products	Items offered for sale by a business	Goods, goods	Customer, Activity, Payment Method, Service, Order, Reviews

Grouping descriptive sentences

Generic data

We want to create a database for the management of purchases/sales between customers and businesses through specific services made available, such as: home delivery, shipping, on-site collection, etc.

Activity Data

The business registers with the system by providing a VAT number, Password, Name, Address, Description, Working hours, telephone number, Link to the owner website, Logo and category (e.g. Clothing, Bar etc...). Businesses make certain types of services available to purchase their products and choose the accepted payment methods. A business can place orders from another business by accessing exclusive discounts thanks to the VAT number.

Service Data

Method of delivery of goods defined by the business: collection of goods on site, shipping, home delivery, rental, commitment.

Payment Type Data

The activities establish which payment methods will be accepted among the various types available: payment on delivery, by card, PayPal, meal vouchers, virtual currency.

Product Data

The products sold are divided into categories and are characterized by: code, price, description, availability and possible discount.

Order Data

An order placed by the customer will be characterized by: date, seller and customer, service and payment method selected by the customer from those available, total price and obviously the list of desired items.

Customer Data

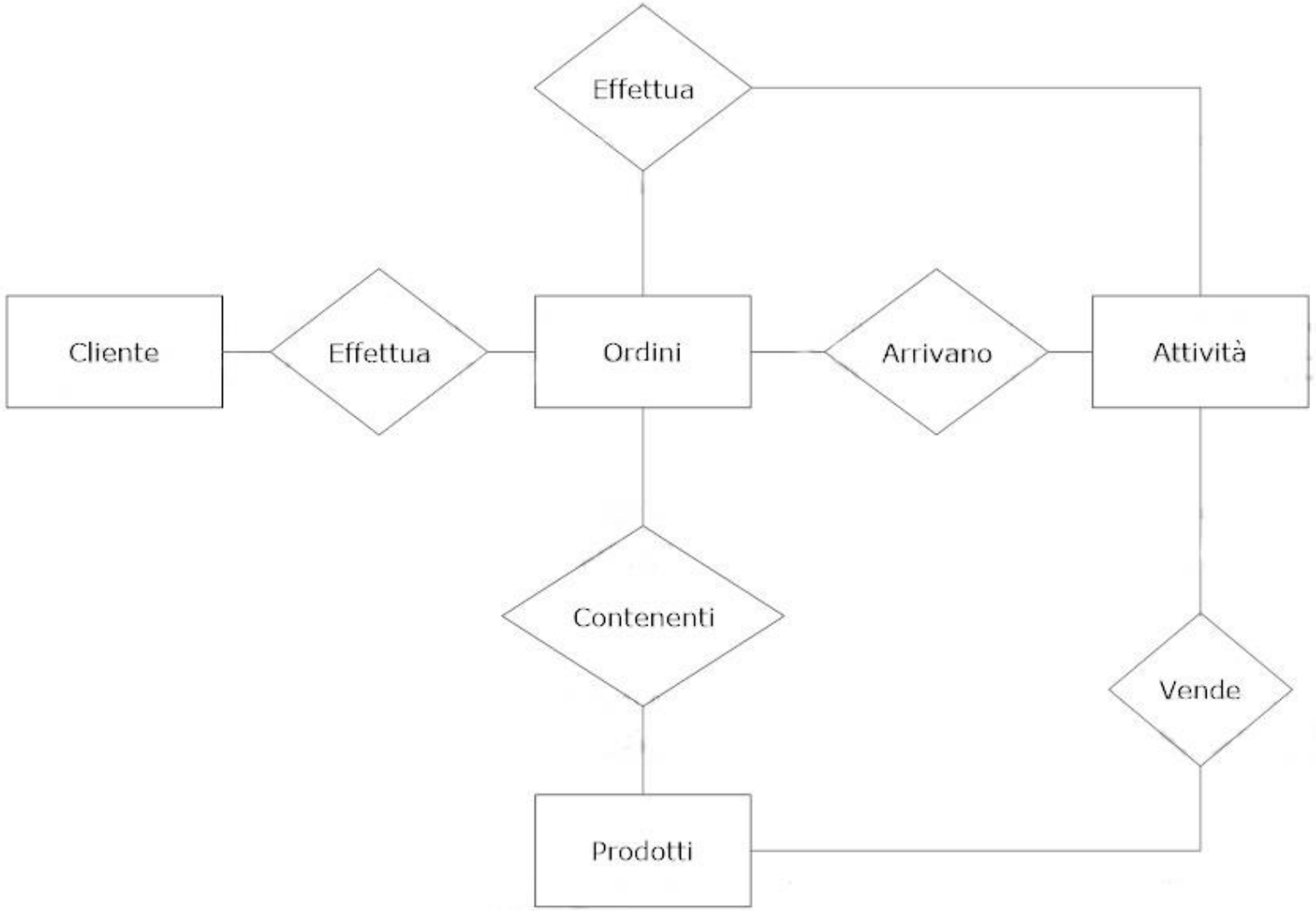
The customer accesses the system via email and password, but will have to enter their name, address and telephone to ensure availability if needed.

Review Data

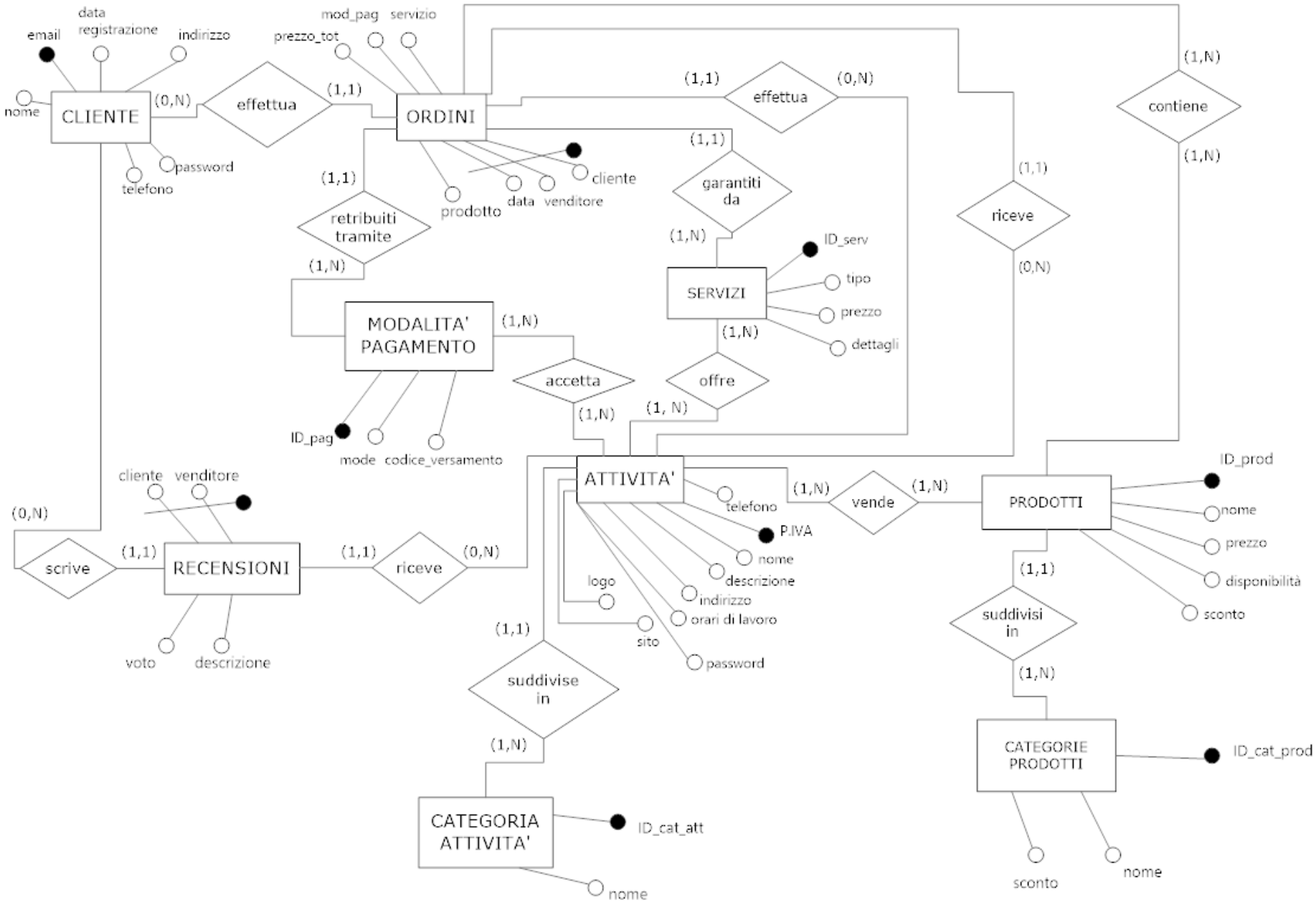
As a feedback system between customer and activity, there is a system for customers to write reviews on the activity which will be made public on the page together with an attached rating on the quality of the purchase.

Conceptual Model Design

Skeleton diagram:



ER scheme:


email
data registrazione
indirizzo
nome
CLIENTE
(0,N)
effettua
(1,1)
mod_pag
servizio
prezzo_tot
ORDINI
password
telefono
(1,1)
retribuiti tramite
(1,N)
prodotto
data
venditore
cliente
(1,1)
effettua
(0,N)
(1,1)
garantiti da
(1,N)
ID_serv
tipo
prezzo
dettagli
SERVIZI
(1,N)
(1,1)
riceve
(0,N)
(1,N)
contiene
(1,N)
MODALITA' PAGAMENTO
(1,N)
accetta
(1,N)
offre
(1, N)
ID_pag
mode
codice_versamento
cliente
venditore
(0,N)
scrive
(1,1)
RECENSIONI
voto
descrizione
(1,1)
riceve
(0,N)
ATTIVITA'
telefono
P.IVA
nome
descrizione
indirizzo
orari di lavoro
password
sito
logo
(1,N)
vende
(1,N)
PRODOTTI
ID_prod
nome
prezzo
disponibilità
sconto
(1,1)
suddivisi in
(1,N)
(1,1)
suddivise in
(1,N)
CATEGORIA ATTIVITA'
ID_cat_att
nome
CATEGORIE PRODOTTI
ID_cat_prod
sconto
nome

Constraints not expressible by the ER scheme

- **Total price of an order:** is obtained as the sum of the individual products participating in the order.
- **Review rating:** the rating is between the values 1 and 10.
- **Activities that order:** it cannot buy products itself.
- **IDs as keys:** these are AUTO-INCREMENT values.
- **Order date:** this is a DATETIME value with TIMESTAMP.
- **Customer registration date:** this is a DATETIME value with TIMESTAMP.
- **Working hours:** "working hours" in a day and "working days" in a week must be distinguished.
- **Address:** is characterized by «Street» and «City».
- **Logo:** it is a link to an image present in a specific HTTP server.

Portion of the volume table

Concept	Guy	Volume
Customer	E	1,000,000
Reviews	E	200,000
Orders	E	2,700,000
Terms of payment	E	6
Services	E	5
Activity	E	800,000
Activity Category	E	54
Products	E	16,000,000
Product Category	E	120
Payments accepted	R	1,000,000
Offered services	R	900,000
Sale of goods	R	30,000,000
Products ordered	R	7,000,000

Operations to be carried out

1. Insertion of new activity (frequency 40 per month)
2. Adding a new customer (frequency 250 per month)
3. Placing an order by a customer (frequency 300 per day)
4. Placement of an order by a business (frequency 100 per month)
5. Writing customer reviews of businesses (80 per day)
6. Viewing the products of a business (freq. 500 per day)
7. Definition of a customer's order history (frequency 300 per day)
8. View history of orders placed (frequency 700 per month)
9. Change to the types of service offered (frequency 30 per year)
10. Change to the types of payment available (frequency 10 per year)
11. Change to the availability of a company's products (freq. 500 per day)

Frequency table

Operation	Description	Frequency	Guy
O1	Insertion of new activity	40/day	I
O2	New customer addition	250/month	I
O3	Placing an order by a customer	300/day	I
O4	Placement of an order by a business	100/month	I
O5	Writing customer reviews of businesses	80/day	I
O6	Viewing a business's products	500/day	I
O8	View history of orders placed	700/month	I
O9	Change to the types of service offered	30/year	I
O10	Change to available payment types	10/year	I
O7	Definition of a customer's order history	300/day	B
O11	Change to the availability of a company's products	500/day	B

Operation scheme

Operation	Entities involved	Relationships involved
O1	Activities, Products	Sale of goods, Services offered, Payments accepted
O2	Customer	
O3	Customer, Order, Products, Activities, Payment Method, Services	Products ordered, Services offered, Payments accepted
O4	Order, Products, Activities, Payment Method, Services	Products ordered, Services offered, Payments accepted
O5	Customers, Reviews	
O6	Categories Products, Products, Activities	Sale of goods
O7	Customer, Order, Products, Activities, Payment Method, Services	Products ordered, Services offered, Payments accepted
O8	Orders	
O9	Services, Activities	Offered services
O10	Payment Methods, Activities	Services offered, Payments accepted
O11	Products	

Logical Model Design

Redundancy analysis

The "product" attribute in Orders is redundant as it can be obtained from Products by navigating through Ordered Products.

Redundant attribute cost evaluation

The "product" attribute will be a varchar(255) and will therefore take up 8 bytes per instance. Having estimated the number of products ordered at 7,000,000, we have 56,000,000 bytes. The operations affected are O7 and O8. The O8 is interactive and therefore needs to be evaluated.

Without redundancy:

Description	E/R	Accesses	Guy
Orders	E	I	L
Ordered Products	R	I	L
Products	E	I	L

There are 3 Reads and 0 Writes, from which: (3*700) = 2100 accesses.

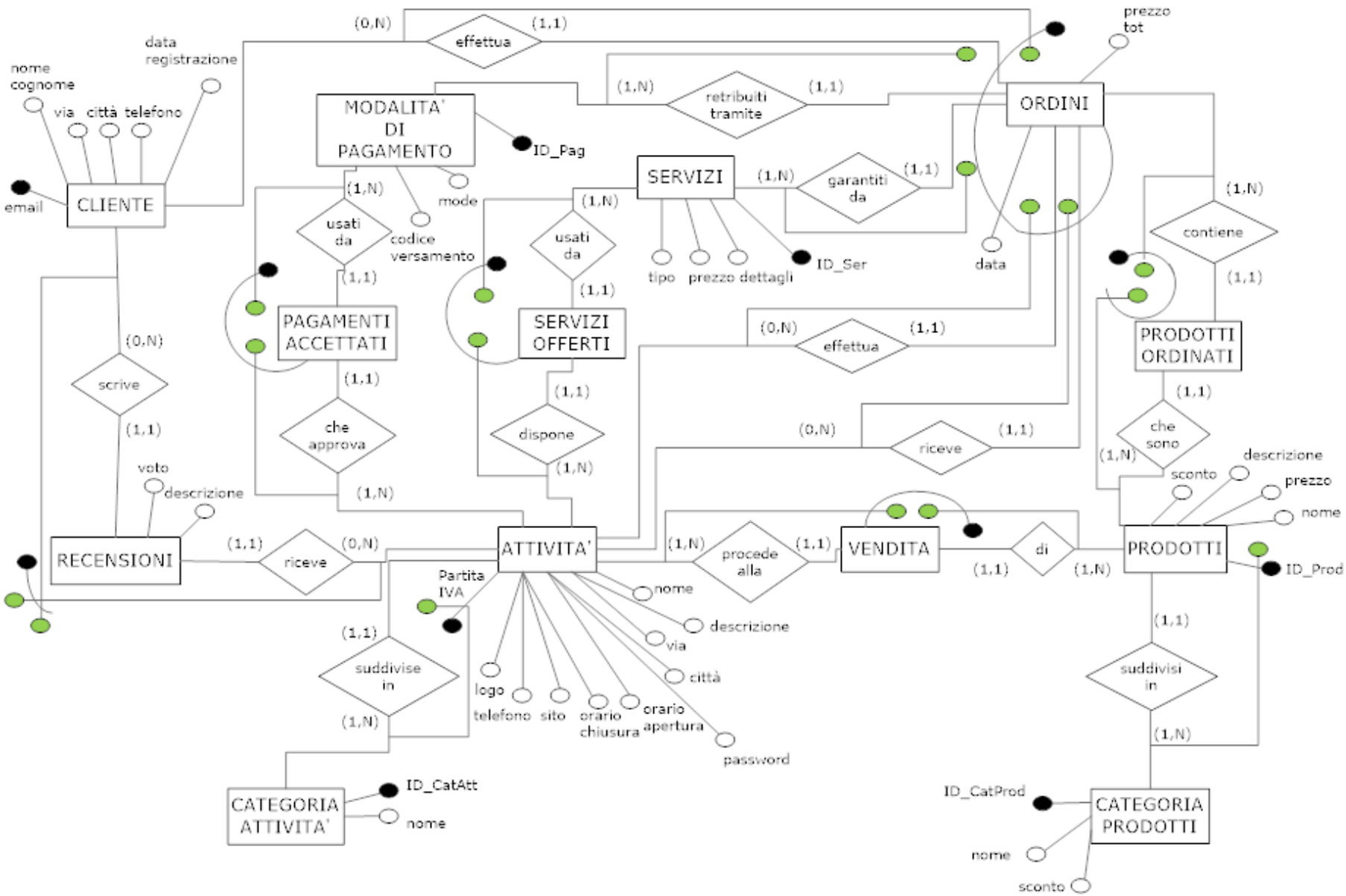
With redundancy:

Description	E/R	Accesses	Guy
Orders	E	I	L

You have 1 Read and 0 Writes, hence: (a*700) = 700 accesses.

There is a saving of 1400 accesses/month compared to 53Mb. It is not convenient to maintain redundancy.

Restructuring of the ER scheme towards the relational model



Customer	(email, name_surname, street, city, telephone, registration_date)
Activity Category	(ID_CatAtt, name)
Activity	(VAT number, Password, name, description, city, street, opening_time, closing_time, working_days, site, logo, telephone, ID_CatAtt)
Reviews	(customer_ID, Att_ID, rating, description)
Services	(ID_Ser, type, price, details)
Payment mode	(Pag_ID, mode, payment_code)
Orders	(date, customer, seller, total_price, page_mode, service)
Product Categories	(ID_CatProd, name, discount)
Products	(Prod_ID, name, price, availability, discount, ID_CatProd)
Ordered Products	(Prod_ID, date, vend, customer)
Offered services	(ID_ser, VAT number)
Payments accepted	(Payment ID, VAT number)
Sale of goods	(VAT number, Product_ID)

Physical Model Design

Below are some SQL statements that will be used to define the relational schema discussed previously:

```
CREATE TABLE activity (
    `VAT number varchar(11) NOT NULL, `Password
    varchar(255) NOT NULL, `Name varchar(255) NOT
    NULL, `Description varchar(255) DEFAULT NULL,

      Čitta ` varchar(255) NOT NULL, `Via
    varchar(255) NOT NULL,
      Opening time          time NOT NULL,
      Closing_time          time NOT NULL,
      Working days              varchar(255) NOT NULL,
      telephone     varchar(10) DEFAULT NULL,
      Yes       varchar(255) DEFAULT NULL,
    `Logo       varchar(255) DEFAULT NULL,
      ÌD_CatAt̀t       int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

- - AUTO_INCREMENT for the activity_category table
- -
ALTER TABLE c̀activity_category
    MODIFY ÌD_CatAt̀t int(11) NOT NULL AUTO_INCREMENT;
- -
- -
- - Limits for the activity table
- -
ALTER TABLE activities
    ADD CONSTRAINT c̀ategoria_activity FOREIGN KEY ( ÌD_CatAt̀t )
REFERENCES category_activity ( ÌD_CatAt̀t ) ON UPDATE CASCADE;
```

```
- - Trigger ordered
- -
CREATE TRIGGER buy_yourself BEFORE INSERT ON orders FOR EACH
ROW
BEGIN
            IF (NEW.customer = NEW.seller) THEN SIGNAL
            SQLSTATE '45000' SET MESSAGE_TEXT ='You cannot
sell products to yourself';
            END IF;
END

CREATE TRIGGER p̀rezzo_totalè AFTER INSERT ON
  he ordered FOR EACH ROW
  BEGIN
            DECLARE total DOUBLE;
            SELECT SUM(price) INTO total FROM products_ordered,
products, orders WHERE orders.date = NEW.date AND
orders.customer = NEW.customer AND orders.seller = NEW.seller AND
orders.date = products_ordered.date AND orders.customer =
products_ordered.customer AND orders.seller = products_ordered.sell
AND products.ID_prod = products_ordered.ID_prod ;



            UPDATE orders SET Price_tot = total WHERE orders.data =
NEW.date AND orders.customer = NEW.customer AND orders.seller
= NEW.seller;

END
```

Below are some PHP and HTML instructions that will be used in defining the web page:

```
<HTML>
<HEAD> <TITLE> At Your Service </TITLE> </HEAD> <BODY>

<H1> <CENTER> At Your Service </CENTER> </H1> <HR>

<H3> <CENTER> Activities Registration Form </CENTER> </H3> <FORM
ACTION="insert_into_attivita.php" METHOD="POST" >
                    VAT number: <input type="text" name="Partita_IVA"><br> Password:
                    <input type="psw" name="pasw"><br>
                    Name: <input type="text" name="name"><br> Description:
                    <input type="text" name="desc"><br> City: <input type="text"
                    name="city" ><br> Street: <input type="text" name="via"><br>

                    Opening time: (hh:mm format) <input type="text" name="ap"><br> Closing time: (hh:mm
                    format)<input type="text" name="closing"><br>
                    Working days: (Mon/Tue/Wed/Thurs/Fri/Sat/Sun format)<input type="text" name="days"><br> Telephone No.: <input
                    type="text" name="tel "><br>
                    Website: <input type="text" name="website"><br> Logo: <input
                    type="text" name="img"><br>
<?php
                    $hostname = "localhost";
                    $username = "root";
                    $password = "";
                    $dbname = "db_project";
                    $conn = mysqli_connect($hostname, $username, $password, $dbname); if(! $conn){

                                        echo "Error connecting to MySQL."; exit();

                    }

                    $sql = "SELECT Name FROM activity_category"; $cont = 0;

                    if ($result = mysqli_query($conn, $sql)) { /* fetch
                    associative array */
                                        while ($row = mysqli_fetch_row($result)) {
                                                            $value=$row[0];
                                        echo '<input type="checkbox" name="category[]" value="'.$row[0].'">'; echo $value."<br>";

                                        }
                    }
                    mysqli_close($conn);
?>
                    <input type="submit" value="Submit"/>
</FORM>
</BODY>
</HTML>
```