Universitatea Politehnica Bucuresti

Proiect Baze de Date

Vanzarea de masini la un dealer auto

Claudiu-Vlad BULIMAC
332AA

Cuprins

1.Introducere

In acest proiect a fost realizata o aplicatie web conceputa pentru a ajuta la gestionarea clientilor, masinilor si a facturilor.

Pentru realizarea aplicatiei, partea functionala a fost realizata in Java iar interfata cu utilizatorul a fost scrisa in fisiere jsp, mediul de dezvolatare folosit fiind IntelliJ IDEA.

Aplicatia a fost conceputa pentru a permite diferite operatii cu entitatile folosite, precum afisarea entitatilor, adaugarea de noi entitati dar si modificarea si stergerea entitatilor deja existente.

Pentru persistarea datelor a fost folosita o baza de date ORALCE.

2.Descrierea aplicatiei

Aplicatia fiind modular s-a incercat pastrarea pe cat posibil a structurii si functionalitatii modulelor, astfel in meniul aferent fiecarui modul se gasesc functiile de vizualizare   si de adaugare.

Functia de vizualizare permite utilizatorului sa vizualizarea tuturor inregistrarilor introduse in aplicatie in modulul respectiv, iar la nivelul fiecarei inregistrari se gaseste butonul “Vizualizare” folosit pentru a implementa functia de vizualizare a unei singure inregistrari dar oferindu-i utilizatorului informatii mai detaliate decat in modul de vizualizare generala.

Functia de adaugare permite utilizatorului adaugarea de noi interogari in baza de date si vizualizarea ulterioara.

Modulul “Clienti” permite vizualizarea, vizualizarea detaliata, editarea si stergerea clientilior.

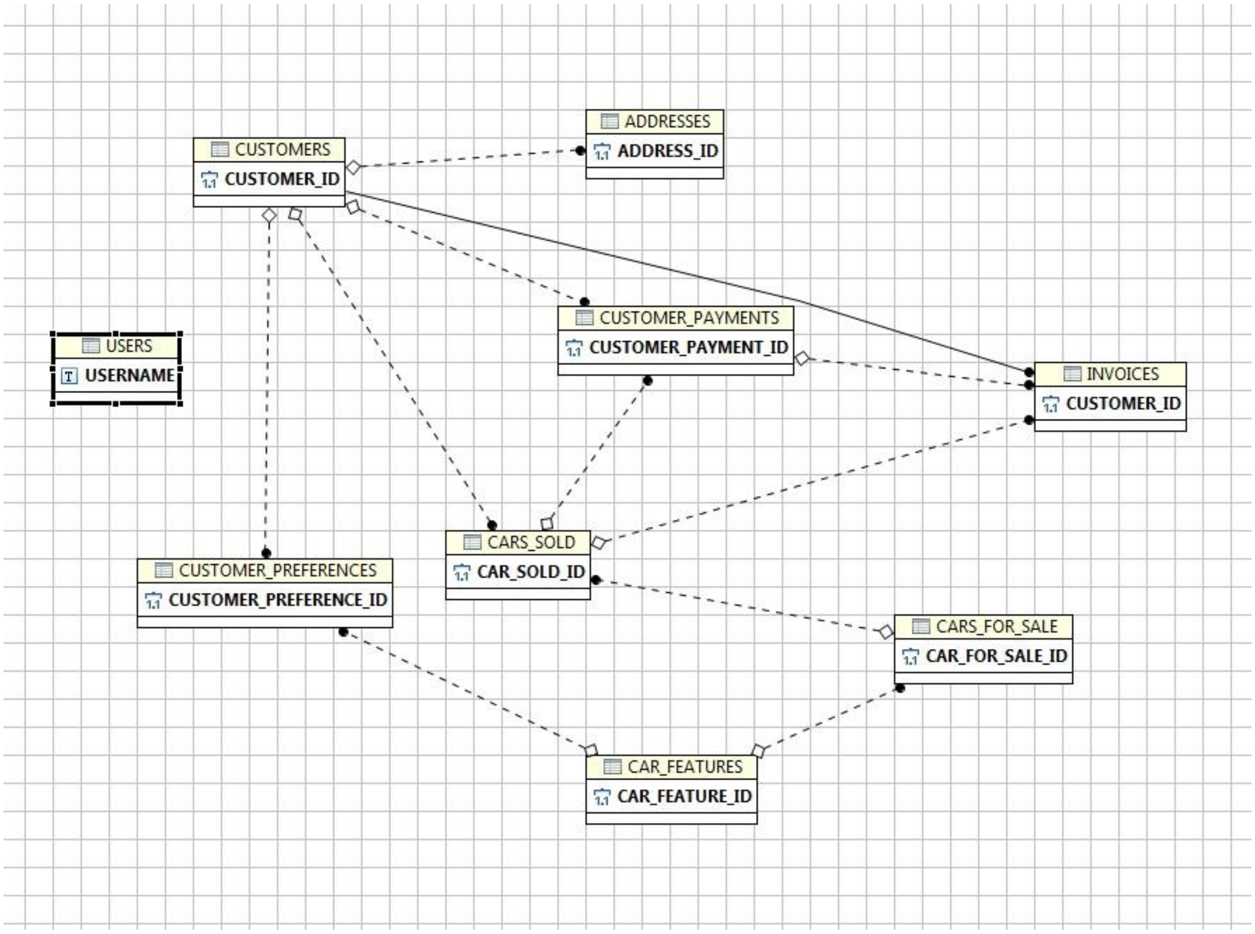
Modulul “Masini” permite vizualizarea, vizualizarea detaliata, editarea si stergerea masinilor disponibile dar si a celor vandute.

Modulul “Facturi” permite vizualizarea combinata a detaliilor clientului, adresa acestuia dar si statusul facturii. Modulul permite si navigarea spre modulul “Clienti” pentru vizualizarea datelor complete ale clientului.

Modulul “Cautare” permite cautarea atat a clientilor cat si a masinilor vandute sau disponibile. Datorita implementarii modulului cautarea este posibila dupa expresii indiferent daca se afla la inceputul, sfarsitul sau in interiorul expresiei cautate. Modulul de cautare nu permite cautarea facturilor de oare ce acest lucru ar putea duce la duplicate in pagina de cautare, incarcarea exagerata a paginii de cautare dar si la incetinirea intoarcerii rezultatelor.

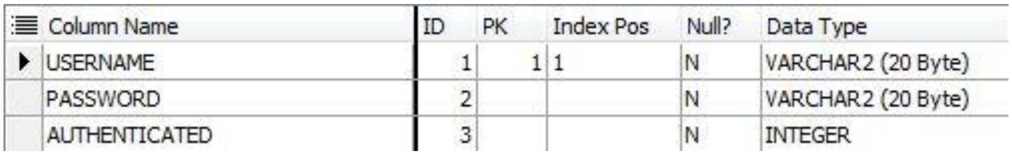
3. Descrierea bazei de date

Pentru persistarea datelor, in dezvoltarea aplicatiei s-a folosit o baza de date ORACLE – OraDB12Home1 si clientul Toad for Oracle 12.6 Freeware.



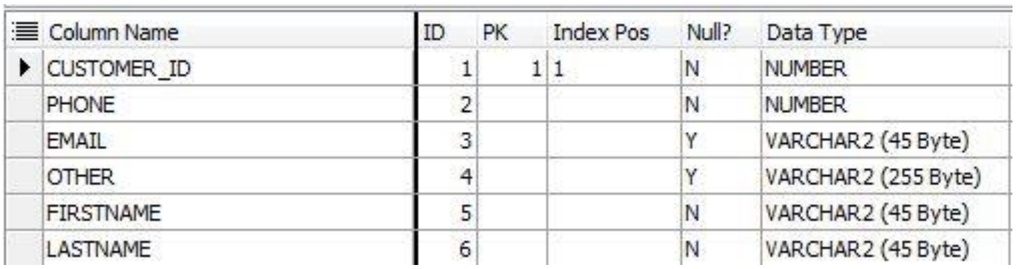
Prezentarea schemei tabelelor:

Users:



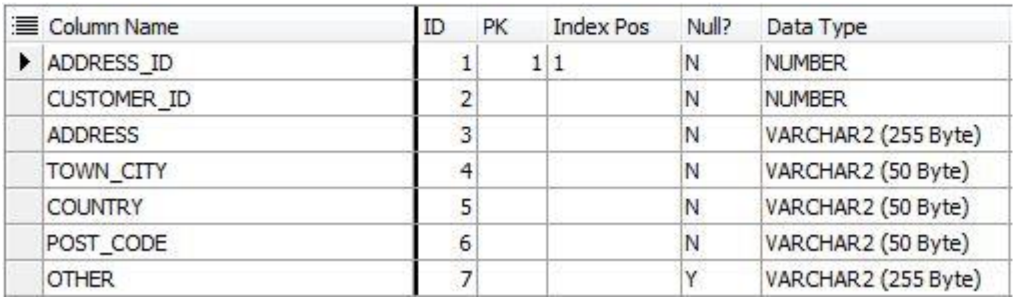
Column Name	ID	PK	Index Pos	Null?	Data Type
USERNAME	1	1	1	N	VARCHAR2 (20 Byte)
PASSWORD	2			N	VARCHAR2 (20 Byte)
AUTHENTICATED	3			N	INTEGER

Customers:



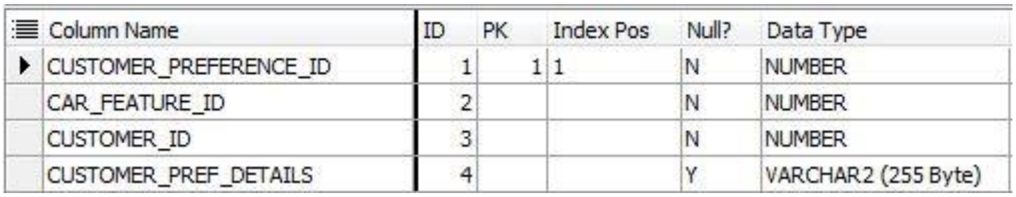
Column Name	ID	PK	Index Pos	Null?	Data Type
CUSTOMER_ID	1	1	1	N	NUMBER
PHONE	2			N	NUMBER
EMAIL	3			Y	VARCHAR2 (45 Byte)
OTHER	4			Y	VARCHAR2 (255 Byte)
FIRSTNAME	5			N	VARCHAR2 (45 Byte)
LASTNAME	6			N	VARCHAR2 (45 Byte)

Addresses:



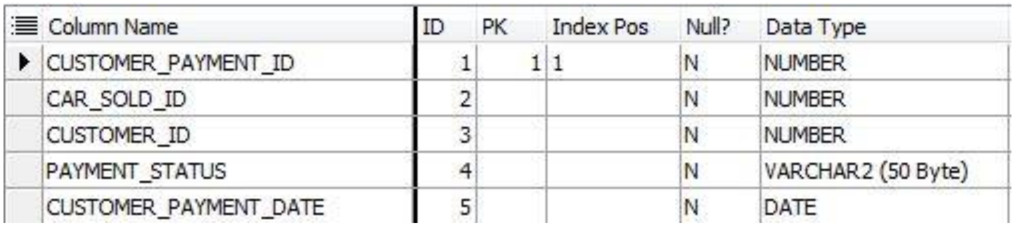
Column Name	ID	PK	Index Pos	Null?	Data Type
ADDRESS_ID	1	1	1	N	NUMBER
CUSTOMER_ID	2			N	NUMBER
ADDRESS	3			N	VARCHAR2 (255 Byte)
TOWN_CITY	4			N	VARCHAR2 (50 Byte)
COUNTRY	5			N	VARCHAR2 (50 Byte)
POST_CODE	6			N	VARCHAR2 (50 Byte)
OTHER	7			Y	VARCHAR2 (255 Byte)

Customer preferences:



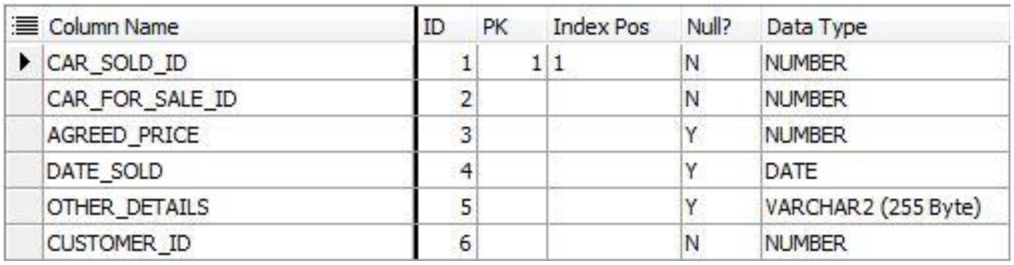
Column Name	ID	PK	Index Pos	Null?	Data Type
CUSTOMER_PREFERENCE_ID	1	1	1	N	NUMBER
CAR_FEATURE_ID	2			N	NUMBER
CUSTOMER_ID	3			N	NUMBER
CUSTOMER_PREF_DETAILS	4			Y	VARCHAR2 (255 Byte)

Customer payments:



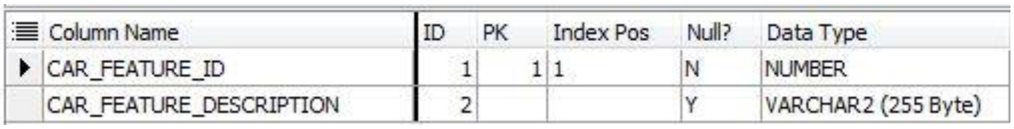
≣	Column Name	ID	PK	Index Pos	Null?	Data Type
▶	CUSTOMER_PAYMENT_ID	1	1	1	N	NUMBER
	CAR_SOLD_ID	2			N	NUMBER
	CUSTOMER_ID	3			N	NUMBER
	PAYMENT_STATUS	4			N	VARCHAR2 (50 Byte)
	CUSTOMER_PAYMENT_DATE	5			N	DATE

Cars Sold:



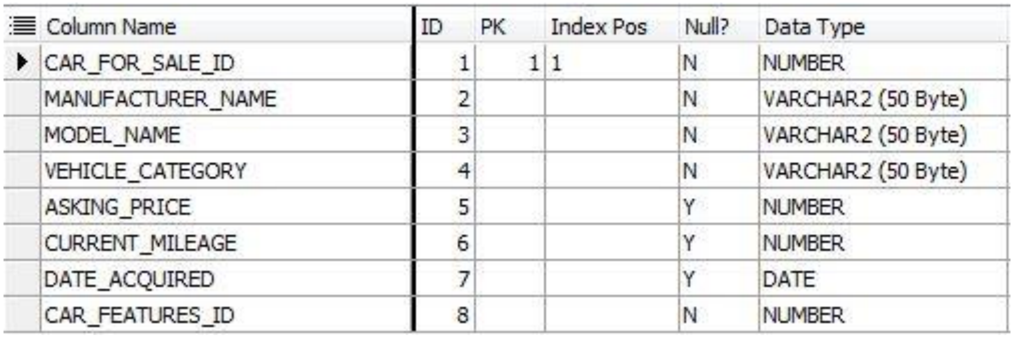
≣	Column Name	ID	PK	Index Pos	Null?	Data Type
▶	CAR_SOLD_ID	1	1	1	N	NUMBER
	CAR_FOR_SALE_ID	2			N	NUMBER
	AGREED_PRICE	3			Y	NUMBER
	DATE_SOLD	4			Y	DATE
	OTHER_DETAILS	5			Y	VARCHAR2 (255 Byte)
	CUSTOMER_ID	6			N	NUMBER

Car features:



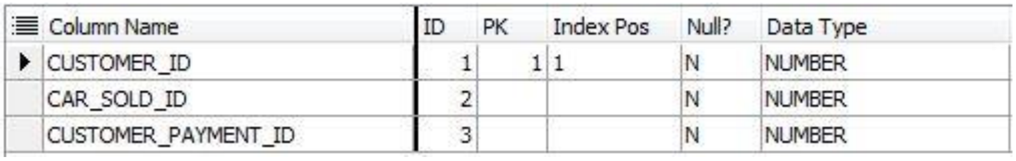
≣	Column Name	ID	PK	Index Pos	Null?	Data Type
▶	CAR_FEATURE_ID	1	1	1	N	NUMBER
	CAR_FEATURE_DESCRIPTION	2			Y	VARCHAR2 (255 Byte)

Cars for sale:



≣	Column Name	ID	PK	Index Pos	Null?	Data Type
▶	CAR_FOR_SALE_ID	1	1	1	N	NUMBER
	MANUFACTURER_NAME	2			N	VARCHAR2 (50 Byte)
	MODEL_NAME	3			N	VARCHAR2 (50 Byte)
	VEHICLE_CATEGORY	4			N	VARCHAR2 (50 Byte)
	ASKING_PRICE	5			Y	NUMBER
	CURRENT_MILEAGE	6			Y	NUMBER
	DATE_ACQUIRED	7			Y	DATE
	CAR_FEATURES_ID	8			N	NUMBER

Invoices:



≣	Column Name	ID	PK	Index Pos	Null?	Data Type
▶	CUSTOMER_ID	1	1	1	N	NUMBER
	CAR_SOLD_ID	2			N	NUMBER
	CUSTOMER_PAYMENT_ID	3			N	NUMBER

4. Query-uri

```
"select * from customers where customer_id='" + id + "'"

"select * from customers ORDER BY customer_id"

"SELECT username, password, AUTHENTICATED FROM users"

"SELECT * from cars_for_sale"

"select * from cars_for_sale where car_for_sale_id =" + id

"select c.customer_id, customer_preference_id, car_feature_id,
customer_pref_details from customers c join customer_preferences
cp on (c.customer_id = cp.customer_id) where c.customer_id="+id

" select c.customer_id, customer_payment_id, payment_status,
customer_payment_date from customers c join customer_payments cp
on (c.customer_id = cp.customer_id) where c.customer_id =" + id

"SELECT c.customer_id, firstname, lastname, phone, email,
c.other, address_id, address, town_city, country, post_code,
a.other as address_other, customer_payment_id, payment_status,
customer_payment_date from customers c join addresses a on
(c.customer_id = a.customer_id) join customer_payments cpay on
(c.customer_id = cpay.customer_id) WHERE c.customer_id =" + id

"select cfs.car_for_sale_id, manufacturer_name,model_name from
cars_for_sale cfs join cars_sold css on (CFS.CAR_FOR_SALE_ID =
CSS.CAR_FOR_SALE_ID WHERE css.customer_id =" + id

"SELECT COUNT(cp.payment_status) as FROM invoices i JOIN
customers c ON (i.customer_id = c.customer_id JOIN cars_sold cs
ON (i.car_sold_id = cs.car_sold_id JOIN customer_payments cp ON
(i.customer_payment_id = cp.customer_payment_id), cars_sold css
JOIN cars_for_sale cfs ON (css.car_for_sale_id =
cfs.car_for_sale_id) WHERE i.customer_id = css.customer_id and
cp.payment_status = '"+ paidStatus +"'"
```

```
"SELECT c.customer_id, c.firstname, c.lastname,
cfs.manufacturer_name, cfs.model_name, cp.payment_status,
cs.agreed_price FROM invoices i JOIN customers c ON
(i.customer_id = c.customer_id JOIN cars_sold cs ON
(i.car_sold_id = cs.car_sold_id JOIN customer_payments cp ON
(i.customer_payment_id = cp.customer_payment_id) cars_sold css
JOIN cars_for_sale cfs ON (css.car_for_sale_id =
cfs.car_for_sale_id) WHERE i.customer_id = css.customer_id ORDER
BY payment_status DESC"

"select cfs.car_for_sale_id, cfs.manufacturer_name,
cfs.model_name, cfs.asking_price from cars_for_sale cfs where
cfs.manufacturer_name IN (select manufacturer_name from
cars_for_sale where LOWER(manufacturer_name) like '%"+
searchText +"%' OR cfs.model_name IN (select model_name from
cars_for_sale where LOWER(model_name) like '%"+ searchText
+"%')"

"select c.customer_id, c.lastname,c.firstname, ad.address_id,
(select a.address from addresses a  where a.customer_id =
c.customer_id) as address from customers c join addresses ad
on(c.customer_id = ad.customer_id) where (LOWER(c.firstname)
like '%"+ searchText +"%') OR (LOWER(c.lastname) like '%"+
searchText +"%')"

"SELECT c.customer_id, firstname, lastname, phone, email,
c.other, address_id, address, town_city, country,
post_code,a.other as address_other, customer_preference_id,
car_feature_id, customer_pref_details, customer_payment_id,
payment_status, customer_payment_date, cs.car_sold_id,
cs.agreed_price, cs.date_sold, cs.OTHER_DETAILS as carsold_other
from customers c join addresses a on (c.customer_id =
a.customer_id) join customer_preferences cp on (c.customer_id =
cp.customer_id) join customer_payments cpay on (c.customer_id =
cpay.customer_id) join cars_sold cs on (c.customer_id =
cs.customer_id) WHERE c.customer_id =" + id

"select CAR_FEATURE_ID, CAR_FEATURE_DESCRIPTION from
car_features cf join cars_for_sale cfs on (CF.CAR_FEATURE_ID =
CFS.CAR_FEATURES_ID) where cfs.car_for_sale_id = " + id
```