

**Department of Computer Science**

Project Report, Fall 2021-22

|  |  |  |  |
| --- | --- | --- | --- |
| Course | Introduction to Database [Section] | Group |  |

**Group Members:**

|  |  |  |
| --- | --- | --- |
| Student ID | Name | Contribution |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Title:**

|  |
| --- |
|  |

**Instructions:**

* Fill up *name, id, section, project title* and *project description*.
* Introduction (4-5 lines).
* Scenario (descriptive). Mention all the **Entities** and **attributes for each Entity** clearly in your Project Scenario.
* Mid term ERD with Finalized ERD mention the improvement issues.
* Normalization up to 3NF for each table.
* Perform DDL and DML in Oracle and attach screenshots.
  + Create all the necessary tables with proper constrains after normalization. Take one screenshot after creating one table with create query and describe query -> continue for rest of the tables.
  + Insert rows (at least 5) for each table. Take one screenshot after inserting into one table with the insert queries and select all query -> continue for rest of the tables.
  + Query (Generate question and provide solution): 2 simple queries with simple conditions, 1 single row function query, 1 multiple row function query, 1 single row sub query, 1 complex sub query, 4 joining queries – equijoin, non-equijoin, outer join, self-join, 1 view, 1 sequence.

1. **Introduction**

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc. Now a days, it is really hard to get some time to go out and get them by ourselves due to busy life style or lots of works. In order to solve this, E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online.

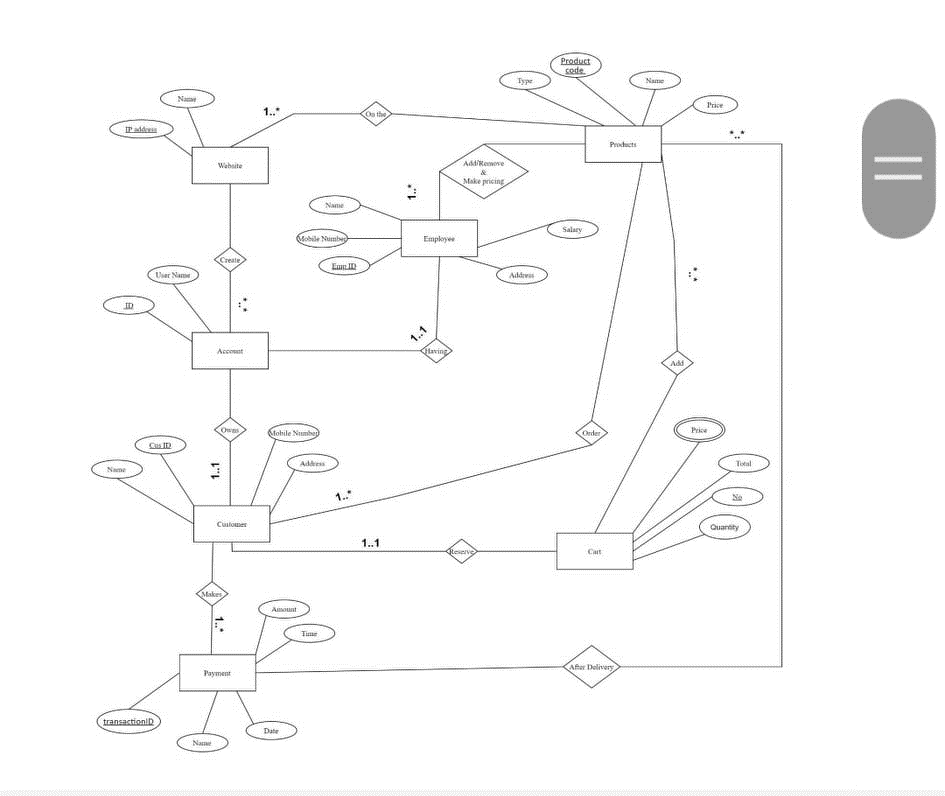
In our project , we will build a complete online shop management system for buyers. This work aims to provide a solution to the problem of online shop management by designing a user friendly computerized system that will be compatible with the existing manual systems. The database to be developed will solve online shop management's problem, thus helping to reduce issues associated with the manual shop management system.

1. **Scenario Description**

This is a database for an online g shop. As it is an online shop, so there has to be a website. So, the website has two attributes which are the website name and its IP address. This website has different accounts for employees and customers. To get access to the account they have to log in to the account. So, to log in to the account, the user has to have a user id and a password. Customers and employees have to have an account in which customers can do shopping and employees can do their daily work. The account has an ID to distinguish and name. Employees have their details in the database which are Id, name, address, salary, mobile number. A customer has their information like ID, name, address and mobile number. There are many products to buy. For the details, products have ID, name, type, amount and price. Customers can add the chosen products to the cart so that they cannot get lost. The cart informs the customer with an ID to track, name, price of each product and the total price. Customers can order a product online. The customer can make payments for the products they have bought through the account. For the payments, it has a payment ID, name, amount and date on which the payment has been made. At last, the bought products are delivered to the customer.

1. **ER Diagram**

**MID-TERM (ER DIAGRAM):**



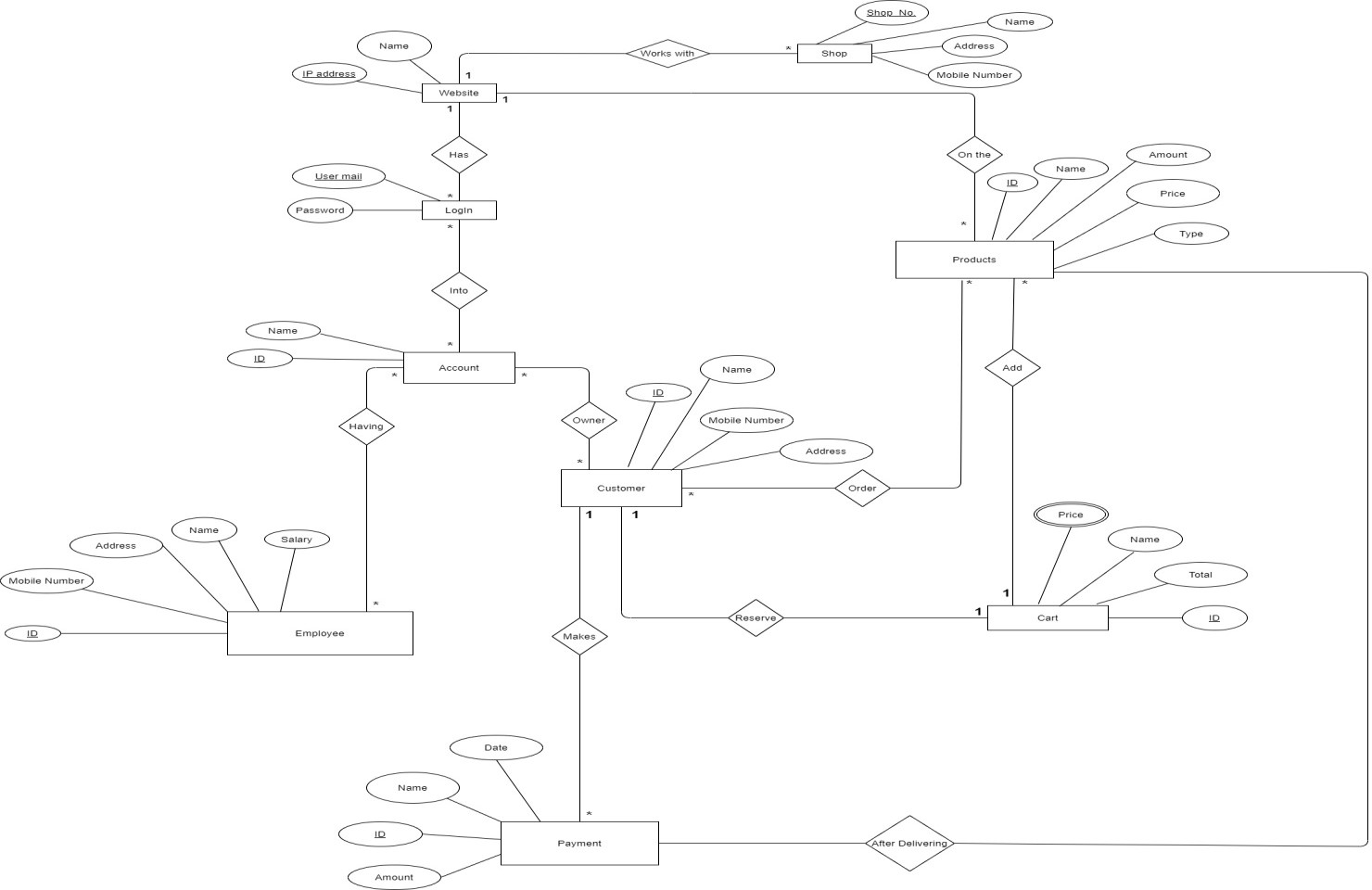
**Major Flaws:**

1. Cardinalities were incorrectly represented
2. Generalization and Specialization were not used

**Update on Fixing Issues:**

1. Cardinalities are fixed and properly used
2. Generalization and specialization are used
3. New Entity is used
4. Some attributes are changed and modified

**FINAL (ER DIAGRAM):**



1. **Normalization**

* **Website works for shops –** ( shop\_no, s\_name, s\_address, m\_number, ip\_address, w\_name)

* + - * **1NF:**

**No multivalued attribute.**

* + - * **2NF:** shop\_no, s\_name, s\_address, m\_number , ip\_address, w\_name

**.** **3NF:**

**No transitive dependency**

shop\_no, s\_name, s\_address, m\_number ip\_address, w\_name

* + - * **Tables from (Shop has website relation)-**

* + - * 1. shop\_no, s\_name, s\_address, m\_number, ip\_address.

**(As Shop has \*-1 relation with Website)**

* + - * 1. ip\_address, w\_name.
* **Website has log-in option-** (ip\_address, w\_name, user\_mail, password)

* + - **1NF:**

No multivalued attribute.

* + - **2NF:** ip\_address, w\_name.

user\_mail, password.

* + - **3NF:**

No transitive dependency.

ip\_address, w\_name. user\_mail, password.

* + - **Tables from (Website has login option)-**

* + - * 1. ip\_address, w\_name.
        2. user\_mail, password, ip\_address.

(As website has 1-\* relation with login option)

* **Log-in into account**- (user\_mail, password, acc\_id, acc\_name)

* + - * **1NF:**

No multivalued attribute.

* + - * **2NF :** user\_mail, password.

acc\_id, acc\_name.

* + - * **3NF :**

No transitive dependency. user\_mail, password. acc\_id, acc\_name.

* + - * **Tables from (Log-in into account)-**

user\_mail, password.

acc\_id, acc\_name.

log\_acc, user\_mail, acc\_id. (As log-in into account has \*-\* relation within it)

* **Employees having accounts**- (e\_id, e\_name, e\_address, e\_number, salary, acc\_id, acc\_name)

* + - * **1NF:**

No multivalued attribute.

* + - * **2NF:**

e\_id, e\_name, e\_address, e\_number, salary. acc\_id, acc\_name.

**3NF:**

No transitive dependency.

e\_id, e\_name, e\_address, e\_number, salary. acc\_id, acc\_name.

* + - * **Tables from (Employees having an account)-**

e\_id, e\_name, e\_address, e\_number, salary.

acc\_id, acc\_name.

emp\_acc, e\_id, acc\_id . (As employee has \*-\* relation with account )

* **Customers having accounts**- (c\_id, c\_name, c\_address, c\_number, acc\_id, acc\_name)

* + - * **1NF:**

No multivalued attribute.

* + - * **2NF:**

c\_id, c\_name, c\_address, c\_number.

acc\_id, acc\_name.

* + - * **3NF:**

No transitive dependency.

c\_id, c\_name, c\_address, c\_number.

acc\_id, acc\_name.

* + - * **Tables from (customers having accounts)-**

c\_id, c\_name, c\_address, c\_number.

acc\_id, acc\_name.

cus\_acc, c\_id, acc\_id. (As customer has \*-\* relation with account )

* **Products on the website-** (p\_id, p\_name, p\_amount, p\_type, p\_price, ip\_address, w\_name)

* + - **1NF:**

No multivalued attribute.

* + - **2NF:**

p\_id, p\_name, p\_amount, p\_type, p\_price.

ip\_address, w\_name.

* + - **3NF:**

No transitive dependency.

p\_id, p\_name, p\_amount, p\_type, p\_price. ip\_address, w\_name.

* + - **Tables from (products on the website)-**

* + - * 1. p\_id, p\_name, p\_amount, p\_type, p\_price, ip\_address.
        2. ip\_address, w\_name. (As products has \*-1 relation with website)

* **Customer order products-** (c\_id, c\_name, c\_address, c\_number, p\_id, p\_name, p\_amount, p\_type, p\_price)

* + - **1NF:**

No multivalued attribute.

* + - **2NF:**

c\_id, c\_name, c\_address, c\_number.

p\_id, p\_name, p\_amount, p\_type, p\_price.

* + - **3NF:**

No transitive dependency.

c\_id, c\_name, c\_address, c\_number.

p\_id, p\_name, p\_amount, p\_type, p\_price.

**Tables from (customer order products)-**

* + - * 1. c\_id, c\_name, c\_address, c\_number.
        2. p\_id, p\_name, p\_amount, p\_type, p\_price.
        3. cus\_pro, c\_id, p\_id . (As customer has \*-\* relation with products)

* **Products get added into the cart-** (p\_id, p\_name, p\_amount, p\_type, p\_price, cart\_id, total, cart\_price)

* + - **1NF:**

Cart price is multivalued attribute.

* + - **2NF:**

p\_id, p\_name, p\_amount, p\_type, p\_price.

cart\_id, total, cart\_price.

* + - **3NF:**

No transitive dependency.

p\_id, p\_name, p\_amount, p\_type, p\_price. cart\_id, total. cart\_id, cart\_price.

* + - **Tables from (products get added into the cart)-**

* + - * 1. p\_id, p\_name, p\_amount, p\_type, p\_price, cart\_id. (As cart has 1-\* relation with products)
        2. cart\_id, total.
        3. cart\_id, cart\_price.

* **Customer reserves cart-** (c\_id, c\_name, c\_address, c\_number, cart\_id, total, cart\_price)

* + - **1NF:**

Cart price is a multivalued attribute.

* + - **2NF:**

c\_id, c\_name, c\_address, c\_number.

cart\_id, total, cart\_price.

* + - **3NF:**

No transitive dependency.

c\_id, c\_name, c\_address, c\_number. cart\_id, total. cart\_id, cart\_price.

* + - **Tables from (customer reserves cart)-**

* + - * 1. c\_id, c\_name, c\_address, c\_number, cart\_id.
        2. cart\_id, total. (As customer has 1-1 relation with cart)
        3. cart\_id, cart\_price.

* **Customer makes payment**- (c\_id, c\_name, c\_address, c\_number, pay\_id, pay\_date, pay\_amount)

* + - **1NF:**

No multivalued attribute.

* + - **2NF:**

c\_id, c\_name, c\_address, c\_number.

pay\_id, pay\_date, pay\_amount.

* + - **3NF:**

No transitive dependency.

c\_id, c\_name, c\_address, c\_number. pay\_id, pay\_date, pay\_amount.

* + - **Tables from (customer makes payment)-**

* + - * 1. c\_id, c\_name, c\_address, c\_number.
        2. pay\_id, pay\_date, pay\_amount, c\_id. (As customer has 1-\* relation with payment)

* **Payment gets done after delivering the product-** (pay\_id, pay\_date, pay\_amount, p\_id, p\_name, p\_amount, p\_type, p\_price)

**1NF:**

No multivalued attribute.

* + - * **2NF:**

pay\_id, pay\_date, pay\_amount.

p\_id, p\_name, p\_amount, p\_type, p\_price.

* + - * **3NF:**

No transitive dependency.

pay\_id, pay\_date, pay\_amount.

p\_id, p\_name, p\_amount, p\_type, p\_price.

* + - * **Tables from (payment gets done after delivering the product)-**

pay\_id, pay\_date, pay\_amount.

p\_id, p\_name, p\_amount, p\_type, p\_price.

pay\_pro, pay\_id, p\_id.

* **Merging all the tables -**

* 1. shop\_no, s\_name, s\_address, s\_number, ip\_address.
  2. ip\_address, w\_name.
  3. ~~ip\_address, w\_name.~~
  4. user\_mail, password, ip\_address.
  5. ~~user\_mail, password.~~
  6. acc\_id, acc\_name.
  7. log\_acc, user\_mail, acc\_id.
  8. e\_id, e\_name, e\_address, e\_number, salary.
  9. ~~acc\_id, acc\_name.~~
  10. emp\_acc, e\_id, acc\_id.
  11. ~~c\_id, c\_name, c\_address, c\_number.~~
  12. ~~acc\_id, acc\_name.~~
  13. cus\_acc, c\_id, acc\_id.
  14. p\_id, p\_name, p\_amount, p\_type, p\_price, ip\_address.
  15. ~~ip\_address, w\_name.~~
  16. ~~c\_id, c\_name, c\_address, c\_number.~~
  17. ~~p\_id, p\_name, p\_amount, p\_type, p\_price.~~
  18. cus\_pro, c\_id, p\_id.
  19. p\_id, p\_name, p\_amount, p\_type, p\_price, cart\_id.
  20. cart\_id, total.
  21. cart\_id, cart\_price.
  22. c\_id, c\_name, c\_address, c\_number, cart\_id.
  23. ~~cart\_id, total.~~
  24. ~~cart\_id, cart\_price.~~
  25. ~~c\_id, c\_name, c\_address, c\_number.~~
  26. pay\_id, pay\_date, pay\_amount, c\_id.
  27. ~~pay\_id, pay\_date, pay\_amount.~~
  28. ~~p\_id, p\_name, p\_amount, p\_type, p\_price.~~
  29. pay\_pro, pay\_id, p\_id.

* **The final tables are-**

1. shop\_no, s\_name, s\_address, s\_number, ip\_address.
2. ip\_address, w\_name.
3. user\_mail, password, ip\_address.
4. acc\_id, acc\_name.
5. log\_acc, user\_mail, acc\_id. (LACC)
6. e\_id, e\_name, e\_address, e\_number, salary.
   1. emp\_acc, e\_id, acc\_id.
   2. cus\_acc, c\_id, acc\_id. (Cuacc)
   3. p\_id, p\_name, p\_amount, p\_type, p\_price, ip\_address.
   4. cus\_pro, c\_id, p\_id. (CP)
   5. p\_id, p\_name, p\_amount, p\_type, p\_price, cart\_id.
   6. cart\_id, total.
   7. cart\_id, cart\_price.
   8. c\_id, c\_name, c\_address, c\_number, cart\_id.
   9. pay\_id, pay\_date, pay\_amount, c\_id.
   10. pay\_pro, pay\_id, p\_id. (Paypro)

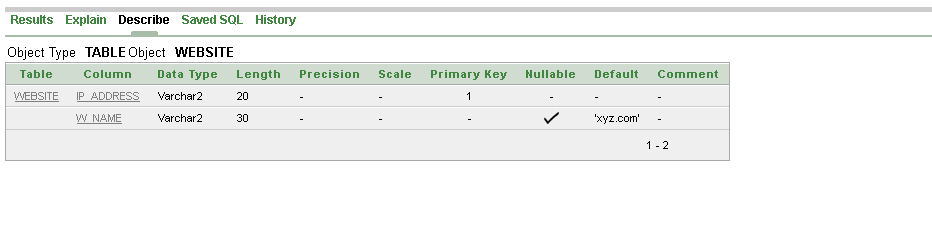
1. **Table Creation:**

**1.**

create table Website ( ip\_address varchar2 (20), w\_name varchar2(30));

alter table Website add constraint Web\_pk primary key(ip\_address)

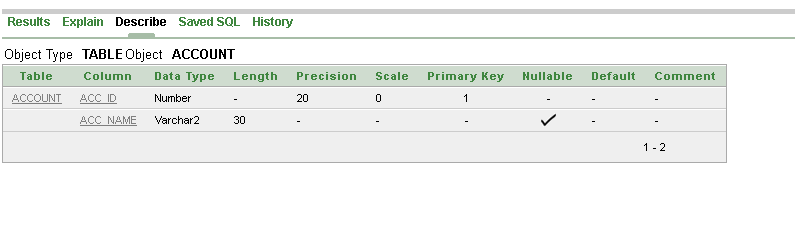
alter table website modify w\_name default 'xyz.com'



**2.**

create table Account ( acc\_id number (20), acc\_name varchar2(30));

alter table Account add constraint Acc\_pk primary key(acc\_id)



**3.**

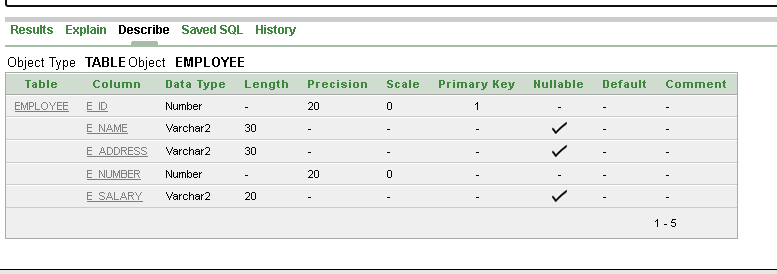
create table Employee ( e\_id number (20), e\_name varchar2(30), e\_address varchar2(30), e\_number number (20), e\_salary varchar2 (20)

);

alter table Employee add constraint Emp\_pk primary key(e\_id)

alter table Employee modify (e\_number not null)

describe Employee



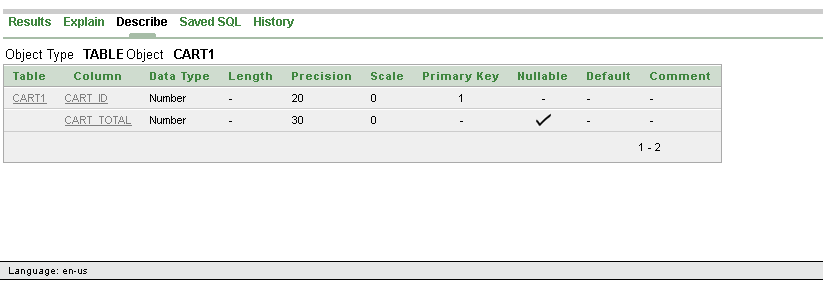
4.

create table Cart1( cart\_id number (20), cart\_total number (30)

);

alter table Cart1 add constraint Cart1\_pk primary key (Cart\_id)

describe cart1



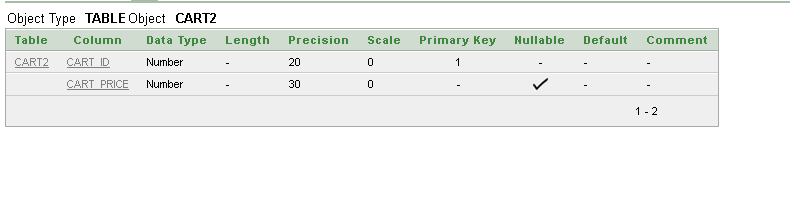
5.

create table Cart2( cart\_id number (20), cart\_price number (30)

);

alter table Cart2 add constraint Cart2\_pk primary key (Cart\_id)

describe cart2



6.

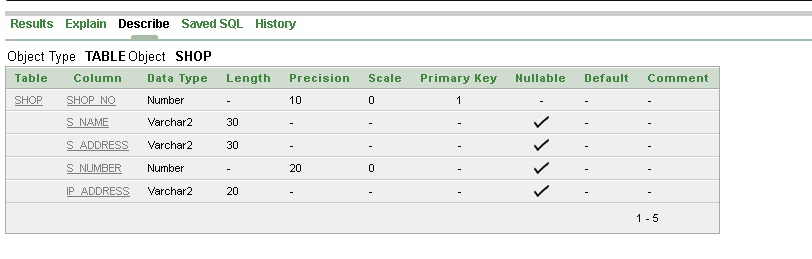
create table Shop ( shop\_no number (10), s\_name varchar2(30), s\_address varchar2(30), s\_number number (20), ip\_address varchar2 (20)

);

alter table Shop add constraint shop\_pk primary key(shop\_no)

alter table Shop add constraint shop\_fk foreign key(ip\_address) references Website(ip\_address)

describe shop



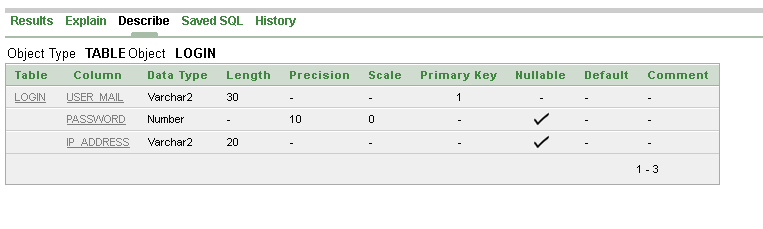
7.

create table Login ( user\_mail varchar2(30)PRIMARY KEY, password number (10), ip\_address varchar2 (20)

)

alter table Login add constraint Login\_fk foreign key(ip\_address) references Website(ip\_address)

describe login



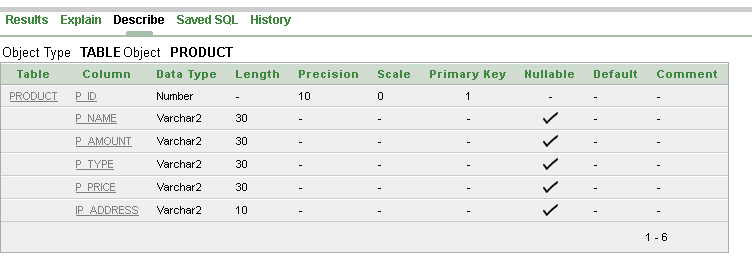
8.

create table Product ( p\_id number (10)PRIMARY KEY, p\_name varchar2 (30), p\_amount varchar2(30), p\_type varchar2(30), p\_price varchar2(30), ip\_address varchar2 (10)

);

alter table Product add constraint product\_fk foreign key(ip\_address) references Website(ip\_address)

describe product



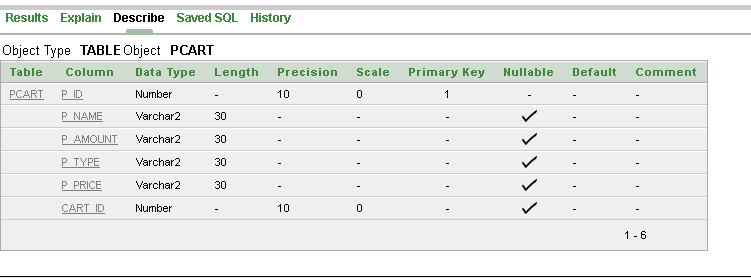
9.

create table Pcart( p\_id number (10)PRIMARY KEY, p\_name varchar2 (30), p\_amount varchar2(30), p\_type varchar2(30), p\_price varchar2(30), cart\_id number (10)

);

alter table Pcart add constraint Pcart\_fk foreign key(cart\_id) references Cart1(cart\_id)

describe Pcart



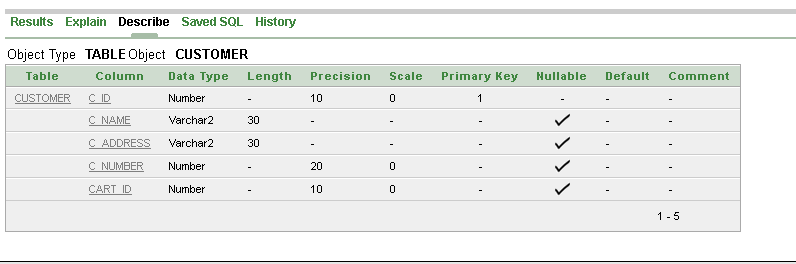
10.

create table Customer ( c\_id number (10)PRIMARY KEY, c\_name varchar2(30), c\_address varchar2(30), c\_number number (20), cart\_id number (10)

);

alter table Customer add constraint Customer\_fk foreign key(cart\_id) references Cart1(cart\_id)

describe Customer



11.

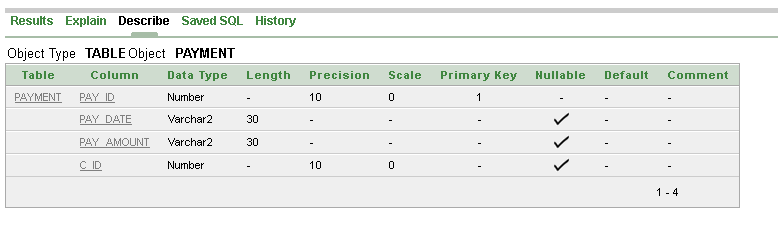
create table Payment ( pay\_id number (10), pay\_date varchar2(30), pay\_amount varchar2(30), c\_id number (10)

);

alter table Payment add constraint Payment\_pk primary key(pay\_id)

alter table Payment add constraint Payment\_fk foreign key(c\_id) references Customer(c\_id)

describe payment



12.

create table Lacc( log\_acc number (10)PRIMARY KEY, user\_mail varchar2(30), acc\_id number (20)

);

alter table Lacc add constraint Lacc\_fk1 foreign key(user\_mail) references Login(user\_mail)

alter table Lacc add constraint Lacc\_fk2 foreign key(acc\_id) references Account(acc\_id)

create sequence La\_seq

start with 1

increment by 2

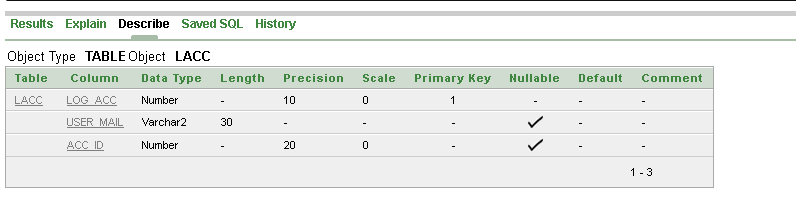
nominvalue

nomaxvalue

nocycle

nocache;

describe Lacc



13.

create table Empacc( emp\_acc varchar2(30) PRIMARY KEY, e\_id number (20), acc\_id number (20)

);

alter table Empacc add constraint Empacc\_fk1 foreign key(e\_id) references Employee(e\_id)

alter table Empacc add constraint Empacc\_fk2 foreign key(acc\_id) references Account(acc\_id)

create sequence Empacc\_seq

start with 20

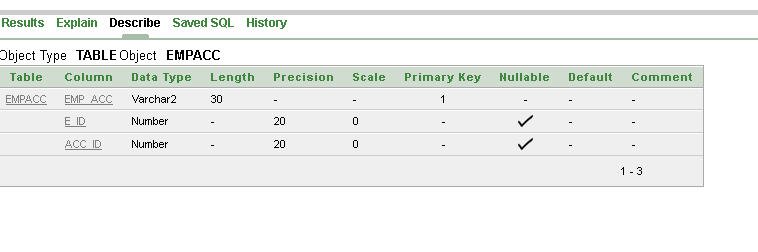
increment by 2

nominvalue

nomaxvalue

nocycle;

describe Empacc



14.

create table Cuacc( cus\_acc varchar2 (30), c\_id number (20), acc\_id number (20)

);

alter table Cuacc add constraint Cuacc\_pk primary key(cus\_acc)

alter table Cuacc add constraint Cuacc\_fk1 foreign key(c\_id) references Customer(c\_id)

alter table Cuacc add constraint Cuacc\_fk2 foreign key(acc\_id) references Account(acc\_id)

create sequence Cuacc\_seq

start with 60

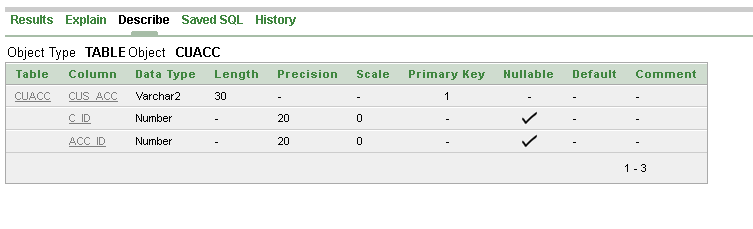
increment by 2

nominvalue

nomaxvalue

nocycle;

describe cuacc



15.

create table Cp ( cus\_pro varchar2(30), c\_id number (10), p\_id number (10)

);

alter table Cp add constraint Cp\_pk primary key(cus\_pro)

alter table Cp add constraint Cp\_fk1 foreign key(c\_id) references Customer(c\_id)

alter table Cp add constraint Cp\_fk2 foreign key(p\_id) references Product(p\_id)

create sequence Cp\_seq start with 80

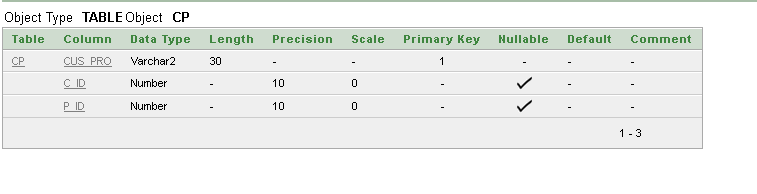
increment by 2

nominvalue

nomaxvalue

nocycle;

describe cp



16.

create table Paypro( pay\_pro varchar2 (20), pay\_id number (20), p\_id number (20)

);

alter table Paypro add constraint Paypro\_pk primary key(pay\_pro)

alter table Paypro add constraint Paypro\_fk1 foreign key(pay\_id) references Payment(pay\_id)

alter table Paypro add constraint Paypro\_fk2 foreign key(p\_id) references Product(p\_id)

create sequence Paypro\_seq

start with 100

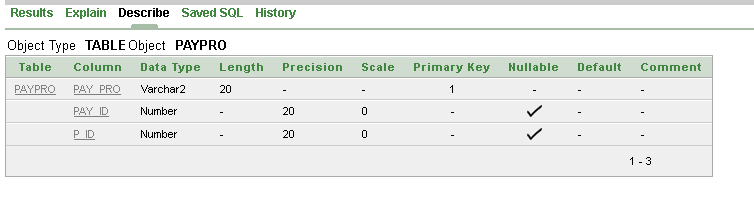
increment by 2

nominvalue

nomaxvalue

nocycle;

describe paypro



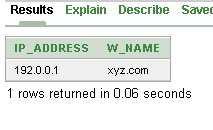
1. Value Insertion Query:

1.

INSERT INTO Website

VALUES ('192.0.0.1','xyz.com');

select \* from website



2.

INSERT INTO Account

VALUES (1234,'Himu');

INSERT INTO Account

VALUES (4567,'Somel');

INSERT INTO Account

VALUES (8912,'Rokon');

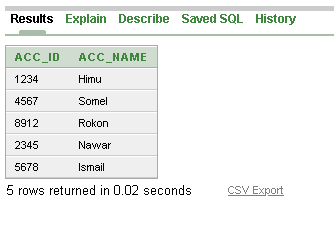
INSERT INTO Account

VALUES (2345,'Nawar');

INSERT INTO Account

VALUES (5678,'Ismail');

select \* from account



3.

INSERT INTO Employee

VALUES (101,'Ismail','Dhaka','153349',5000);

INSERT INTO Employee

VALUES (102,'Faiza','Cumilla','112233',6000);

INSERT INTO Employee

VALUES (103,'Sadia','Rajshahi','646587',7000);

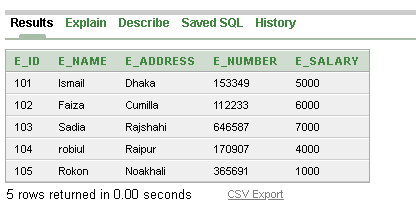
INSERT INTO Employee

VALUES (104,'robiul','Raipur','170907',4000);

INSERT INTO Employee

VALUES (105,'Rokon','Noakhali','365691',1000);

select \* from employee



4.

INSERT INTO Cart1

VALUES (201,10);

INSERT INTO Cart1

VALUES (202,20);

INSERT INTO Cart1

VALUES (203,30);

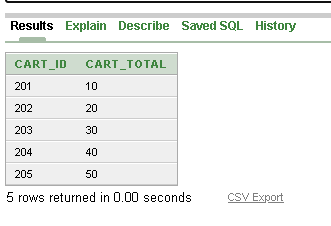
INSERT INTO Cart1

VALUES (204,40);

INSERT INTO Cart1

VALUES (205,50);

select \* from cart1



5.

INSERT INTO Cart2

VALUES (301,1);

INSERT INTO Cart2

VALUES (302,2);

INSERT INTO Cart2

VALUES (303,3);

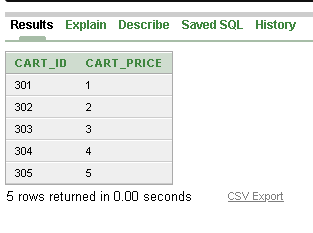
INSERT INTO Cart2

VALUES (304,4);

INSERT INTO Cart2

VALUES (305,5);

select \* from cart2



6.

INSERT INTO Shop (ip\_address,shop\_no,s\_name,s\_address,s\_number)

VALUES ('192.0.0.1',1,'Nora Store','Dhaka',1720504996);

INSERT INTO Shop (ip\_address,shop\_no,s\_name,s\_address,s\_number)

VALUES ('192.0.0.1',2,'Cutu Supershop','Banasree',1620504994);

INSERT INTO Shop (ip\_address,shop\_no,s\_name,s\_address,s\_number)

VALUES ('192.0.0.1',3,'Robi Cosmetics','Raipur',1727061644);

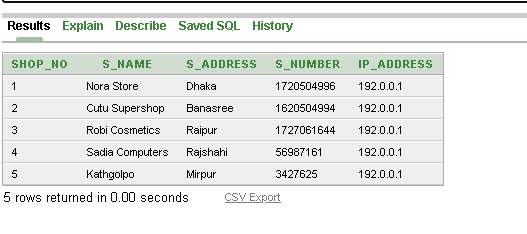
INSERT INTO Shop (ip\_address,shop\_no,s\_name,s\_address,s\_number)

VALUES ('192.0.0.1',4,'Sadia Computers','Rajshahi',56987161);

INSERT INTO Shop (ip\_address,shop\_no,s\_name,s\_address,s\_number)

VALUES ('192.0.0.1',5,'Kathgolpo','Mirpur',3427625);

select \* from shop



7.

INSERT INTO Login (ip\_address,user\_mail,password)

VALUES ('192.0.0.1','somel@gmail.com',47527);

INSERT INTO Login (ip\_address,user\_mail,password)

VALUES ('192.0.0.1','rokon@hotmail.com',75275);

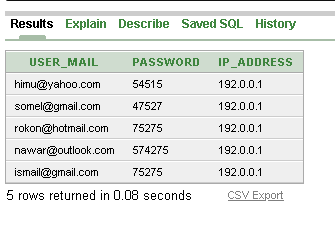
INSERT INTO Login (ip\_address,user\_mail,password)

VALUES ('192.0.0.1','nawar@outlook.com',574275);

INSERT INTO Login (ip\_address,user\_mail,password)

VALUES ('192.0.0.1','ismail@gmail.com',75275);

select \* from login



8.

INSERT INTO Lacc (log\_acc,user\_mail,acc\_id)

VALUES (1001,'himu@yahoo.com',1234);

INSERT INTO Lacc (log\_acc,user\_mail,acc\_id)

VALUES (1002,'somel@gmail.com',4567);

INSERT INTO Lacc (log\_acc,user\_mail,acc\_id)

VALUES (1003,'rokon@hotmail.com',8912);

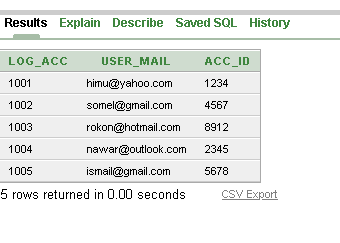
INSERT INTO Lacc (log\_acc,user\_mail,acc\_id)

VALUES (1004,'nawar@outlook.com',2345);

INSERT INTO Lacc(log\_acc,user\_mail,acc\_id)

VALUES (1005,'ismail@gmail.com',5678);

select \* from lacc



9.

INSERT INTO Empacc(emp\_acc,e\_id,acc\_id)

VALUES (2001,101,1234);

INSERT INTO Empacc(emp\_acc,e\_id,acc\_id)

VALUES (2002,102,4567);

INSERT INTO Empacc(emp\_acc,e\_id,acc\_id)

VALUES (2003,103,8912);

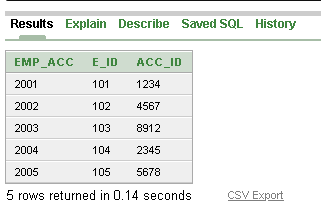
INSERT INTO Empacc(emp\_acc,e\_id,acc\_id)

VALUES (2004,104,2345);

INSERT INTO Empacc(emp\_acc,e\_id,acc\_id)

VALUES (2005,105,5678);

select \* from empacc



10.

INSERT INTO Product (ip\_address, p\_id, p\_name, p\_amount, p\_type, p\_price) VALUES ('192.0.0.1', 101, 'paper', '100', 'solid', '500');

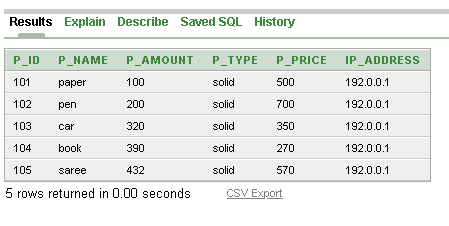
INSERT INTO Product (ip\_address, p\_id, p\_name, p\_amount, p\_type, p\_price) VALUES ('192.0.0.1', 102,'pen', '200', 'solid', '700');

INSERT INTO Product (ip\_address, p\_id, p\_name, p\_amount, p\_type, p\_price) VALUES ('192.0.0.1', 103, 'car', '320' ,'solid', '350');

INSERT INTO Product (ip\_address, p\_id, p\_name, p\_amount, p\_type, p\_price) VALUES ('192.0.0.1', 104, 'book', '390','solid', '270');

INSERT INTO Product (ip\_address, p\_id, p\_name, p\_amount, p\_type, p\_price) VALUES ('192.0.0.1', 105, 'saree', '432','solid', '570');

select \* from product



11.

INSERT INTO Pcart (p\_id, p\_name, p\_amount, p\_type, p\_price, cart\_id) VALUES (101, 'paper', '100', 'solid', '500', 201);

INSERT INTO Pcart ( p\_id, p\_name, p\_amount, p\_type, p\_price , cart\_id)

VALUES (102, 'lipstick ', '200', 'solid', '700' ,202);

INSERT INTO Pcart ( p\_id, p\_name, p\_amount, p\_type, p\_price , cart\_id)

VALUES (103, 'car', '320', 'solid', '350', 203);

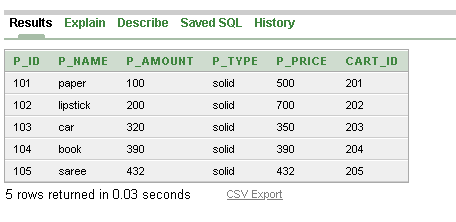
INSERT INTO Pcart ( p\_id, p\_name, p\_amount, p\_type, p\_price , cart\_id)

VALUES (104, 'book', '390', 'solid', '390', 204);

INSERT INTO Pcart ( p\_id, p\_name, p\_amount, p\_type, p\_price , cart\_id)

VALUES (105, 'saree', '432', 'solid', '432', 205);

select \* from pcart



12.

INSERT INTO Customer (cart\_id , c\_id, c\_name, c\_address, c\_number)

VALUES (201, 789456, 'faiza20', 'Bashundhara',000345)

INSERT INTO Customer (cart\_id , c\_id, c\_name, c\_address, c\_number)

VALUES (202, 889466, 'hemal67', 'Banasree',001346)

INSERT INTO Customer (cart\_id , c\_id, c\_name, c\_address, c\_number)

VALUES (203, 989476, 'rabi23', 'Uttara',002347)

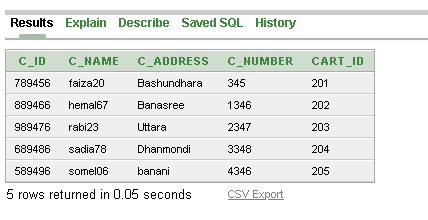
INSERT INTO Customer (cart\_id , c\_id, c\_name, c\_address, c\_number)

VALUES (204, 689486, 'sadia78', 'Dhanmondi',003348)

INSERT INTO Customer (cart\_id , c\_id, c\_name, c\_address, c\_number)

VALUES (205, 589496, 'somel06', 'banani',004346)

select \* from customer



13.

INSERT INTO Payment (c\_id ,pay\_date, pay\_id, pay\_amount)

VALUES ( 789456, '19.04.2019', 601, '200$')

INSERT INTO Payment (c\_id ,pay\_date, pay\_id, pay\_amount)

VALUES ( 889466, '24.05.2020', 701, '250$')

INSERT INTO Payment (c\_id ,pay\_date, pay\_id, pay\_amount)

VALUES ( 989476, '06.08.2020', 801, '300$')

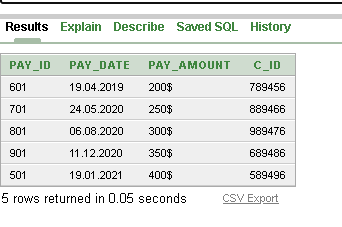
INSERT INTO Payment (c\_id ,pay\_date, pay\_id, pay\_amount)

VALUES ( 689486, '11.12.2020', 901, '350$')

INSERT INTO Payment (c\_id ,pay\_date, pay\_id, pay\_amount)

VALUES ( 589496, '19.01.2021', 501, '400$')

select \* from payment



14.

INSERT INTO Cuacc (c\_id, acc\_id, cus\_acc)

VALUES ( 789456, 1234, 'fai0002020')

INSERT INTO Cuacc (c\_id, acc\_id, cus\_acc)

VALUES ( 889466, 4567, 'hem0002020')

INSERT INTO Cuacc (c\_id, acc\_id, cus\_acc)

VALUES ( 989476, 8912, 'robi0002020')

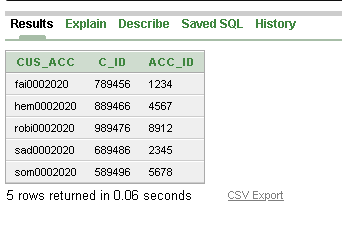
INSERT INTO Cuacc (c\_id, acc\_id, cus\_acc)

VALUES ( 689486, 2345, 'sad0002020')

INSERT INTO Cuacc (c\_id, acc\_id, cus\_acc)

VALUES ( 589496, 5678, 'som0002020')

select \* from cuacc



15.

INSERT INTO Cp (cus\_pro, c\_id, p\_id)

VALUES ('Farzana faiza', 789456, 101)

INSERT INTO Cp (cus\_pro, c\_id, p\_id)

VALUES ('Hemal karmakar', 889466, 102)

INSERT INTO Cp (cus\_pro, c\_id, p\_id)

VALUES ('Robiul islam', 989476, 103)

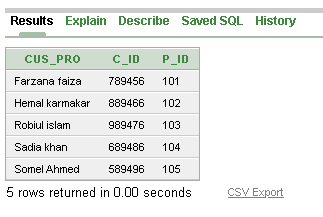
INSERT INTO Cp (cus\_pro, c\_id, p\_id)

VALUES ('Sadia khan', 689486, 104)

INSERT INTO Cp (cus\_pro, c\_id, p\_id)

VALUES ('Somel Ahmed', 589496, 105)

select \* from cp



16.

INSERT INTO Paypro (pay\_pro, pay\_id, p\_id)

VALUES ('RCOOFAZVISA', 601, 101)

INSERT INTO Paypro (pay\_pro, pay\_id, p\_id)

VALUES ('RCOOHEMCARD', 701, 102)

INSERT INTO Paypro (pay\_pro, pay\_id, p\_id)

VALUES ('RCOOROBCZ', 801, 103)

INSERT INTO Paypro (pay\_pro, pay\_id, p\_id)

VALUES ('RCOOSADCARD', 901, 104)

INSERT INTO Paypro (pay\_pro, pay\_id, p\_id)

VALUES ('RCOOPEOVISA', 501, 104)

select \* from paypro

