

Department of Computer Science and Engineering

B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Lab work entitled “Database Management Systems (22CS3PCDBM)” conducted by **SANTOSH H JAMBAGI (1BM22CS244)**, who is bonafide student at **B.M.S.College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** during the academic year 2023-24. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (23CS3PCDBM) work prescribed for the said degree.

Vikranth B M
Assistant Professor
Course Instructor
Database Management Systems

Dr. Jyoti S Nayak
Head of the Department
Dept. of CSE

1. Insurance Database

PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below:

PERSON (driver_id: String, name: String, address: String)

CAR (reg_num: String, model: String, year: int)

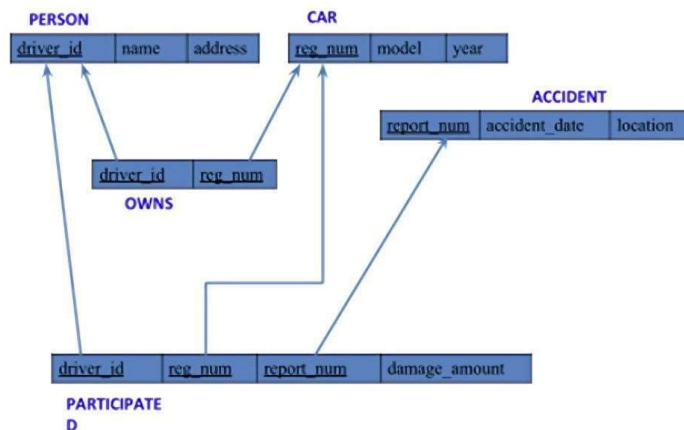
ACCIDENT (report_num: int, accident_date: date, location: String)

OWNS (driver_id: String, reg_num: String)

PARTICIPATED (driver_id: String, reg_num: String, report_num: int, damage_amount: int)

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- Display Accident date and location.
- Update the damage amount to 25000 for the car with a specific reg_num (example 'KA053408') for which the accident report number was 12.
- Add a new accident to the database.
- Display driver id who did accident with damage amount greater than or equal to Rs.25000.

Schema Diagram:



Creating Database and Table:

```
Create database insurance_244;
```

```
Use insurance_244;
```

```

insert into car values("KA052250","Indica", "1990");
insert into car values("KA031181","Lancer", "1957");
insert into car values("KA095477","Toyota", "1998");
insert into car values("KA053408","Honda", "2008");
insert into car values("KA041702","Audi", "2005");
select * from car;

```

reg_num	model	year
KA031181	Lancer	1957
KA041702	Audi	2005
KA052250	Indica	1990
KA053408	Honda	2008
KA095477	Toyota	1998
NULL	NULL	NULL

```

insert into owns values("A01","KA052250");
insert into owns values("A02","KA031181");
insert into owns values("A03","KA095477");
insert into owns values("A04","KA053408");
insert into owns values("A05","KA041702");
select * from owns;

```

driver_id	reg_num
A02	KA031181
A05	KA041702
A01	KA052250
A04	KA053408
A03	KA095477
NULL	NULL

```

insert into accident values(11,'2003-01-01',"Mysore Road");
insert into accident values(12,'2004-02-02',"South end Circle");
insert into accident values(13,'2003-01-21',"Bull temple Road");
insert into accident values(14,'2008-02-17',"Mysore Road");
insert into accident values(15,'2004-03-05',"Kanakpura Road");
select * from accident;

```

report_num	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end Circle
13	2003-01-21	Bull temple Road
14	2008-02-17	Mysore Road
15	2004-03-05	Kanakpura Road
NULL	NULL	NULL

```

insert into participated values("A01","KA052250",11,10000);
insert into participated values("A02","KA053408",12,50000);
insert into participated values("A03","KA095477",13,25000);
insert into participated values("A04","KA031181",14,3000);
insert into participated values("A05","KA041702",15,5000);
select * from participated;

```

driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

Queries :

I. Display accident date and location .
select accident_date, location from accident;

accident_date	location
2003-01-01	Mysore Road
2004-02-02	South end Circle
2003-01-21	Bull temple Road
2008-02-17	Mysore Road
2004-03-05	Kanakpura Road

II. Add a new accident to the database.
insert into accident values(16,'2008-03-08',"Domlur");
select * from accident;

report_num	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end Circle
13	2003-01-21	Bull temple Road
14	2008-02-17	Mysore Road
15	2004-03-05	Kanakpura Road
16	2008-03-08	Domlur
NULL	NULL	NULL

III. Display driver id who did accident with damage amount greater than or equal to rs.25000.
select driver_id from participated where damage_amount >= 25000;

driver_id
A02
A03

2. More Queries on Insurance Database

PROGRAM 2. More Queries on Insurance Database

PERSON (driver_id: String, name: String, address: String)

CAR (reg_num: String, model: String, year: int)

ACCIDENT (report_num: int, accident_date: date, location: String)

OWNS (driver_id: String, reg_num: String)

PARTICIPATED (driver_id: String, reg_num: String, report_num: int, damage_amount: int)

Create the above tables by properly specifying the primary keys and the foreign keys as done in “Program 1” week’s lab and Enter at least five tuples for each relation.

- i. Display the entire CAR relation in the ascending order of manufacturing year.
- ii. Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved. iii. Find the total number of people who owned cars that involved in accidents in 2008.
- iv. List the Entire Participated Relation in the Descending Order of Damage Amount. Find the Average Damage Amount.
- v. Delete the Tuple Whose Damage Amount is below the Average Damage Amount.
- vi. List the Name of Drivers Whose Damage is Greater than The Average Damage Amount. vii. Find Maximum Damage Amount.

Creating database and table:

Database insurance_141 and tables as per schema were created in the previous lab and it is as shown in the previous experiment.

Queries :

- i. **Display the entire CAR relation in the ascending order of manufacturing year.**

select * from car order by year asc;

reg_num	model	year
KA031181	Lancer	1957
KA052250	Indica	1990
KA095477	Toyota	1998
KA041702	Audi	2005
KA053408	Honda	2008
HULL	HULL	HULL

- ii. **Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.**

select count(report_num) from car c, participated p
where c.reg_num=p.reg_num and c.model='Lancer';

count(report_num)
1

- iii. **Find the total number of people who owned cars that were involved in accidents in 2008.**

select count(distinct driver_id) CNT from participated a, accident b
where a.report_num=b.report_num and b.accident_date like '%2008%';

CNT
1

3. Bank Database

PROGRAM 3: Bank Database

Branch (branch-name: String, branch-city: String, assets: real)

BankAccount(accno: int, branch-name: String, balance: real)

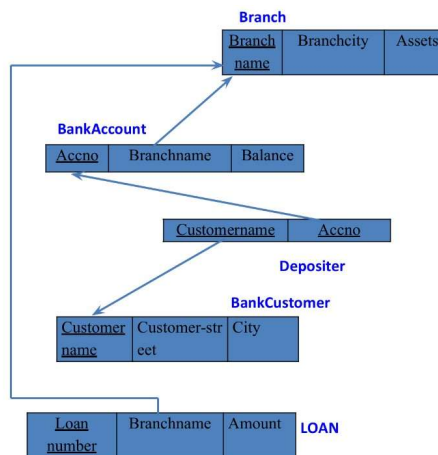
BankCustomer (customer-name: String, customer-street: String, customer-city: String)

Depositer(customer-name: String, accno: int)

LOAN (loan-number: int, branch-name: String, amount: real)

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.
- Find all the customers who have at least two accounts at the same branch (ex. SBI_ResidencyRoad).
- Create A View Which Gives Each Branch the Sum of The Amount of All The Loans At The Branch.

Schema Diagram :



Creating Database and Table:

```
create database bank_244;  
use bank_244;
```

```
Create table branch(  
Branch_name varchar(30),  
Branch_city varchar(25),  
assets int,  
PRIMARY KEY (Branch_name)  
);
```

```
Create table BankAccount(  
Accno int,  
Branch_name varchar(30),
```

```

insert into BankCustomer values("Avinash","Bull_Temple_Road","Bangalore");
insert into BankCustomer values("Dinesh","Bannerghatta_Road","Bangalore");
insert into BankCustomer values("Mohan","NationalCollege_Road","Bangalore");
insert into BankCustomer values("Nikil","Akbar_Road","Delhi");
insert into BankCustomer values("Ravi","Prithviraj_Road","Delhi");
select * from BankCustomer;

```

Customername	Customer_street	CustomerCity
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannerghatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

```

insert into Depositer values("Avinash",1);
insert into Depositer values("Dinesh",2);
insert into Depositer values("Nikil",4);
insert into Depositer values("Ravi",5);
insert into Depositer values("Avinash",8);
insert into Depositer values("Nikil",9);
insert into Depositer values("Dinesh",10);
insert into Depositer values("Nikil",11);
select * from Depositer;

```

Customername	Accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11
NULL	NULL

```

insert into Loan values(1,"SBI_Chamrajpet",1000);
insert into Loan values(2,"SBI_ResidencyRoad",2000);
insert into Loan values(3,"SBI_ShivajiRoad",3000);
insert into Loan values(4,"SBI_ParlimentRoad",4000);
insert into Loan values(5,"SBI_Jantarmantar",5000);
select * from Loan;

```

Loan_number	Branch_name	Amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

Queries :

iii. Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.

```
select branch_name, assets as assets_in_lakhs from branch;
```

Branch_name	assets_in_lakhs
SBI_Chamrajpet	0.5000 lakhs
SBI_Jantarmantar	0.2000 lakhs
SBI_ParlimentRoad	0.1000 lakhs
SBI_ResidencyRoad	0.1000 lakhs
SBI_ShivajiRoad	0.2000 lakhs

iv. Find all the customers who have at least two accounts at the same branch (ex.SBI_ResidencyRoad).

```

select d.Customername from Depositer d, BankAccount b where
b.Branch_name='SBI_ResidencyRoad' and d.Accno=b.Accno group by
d.Customername having count(d.Accno)>=2;

```

Customername
Dinesh

4. More Queries on Bank Database

PROGRAM 4: More Queries on Bank Database

Branch (branch-name: String, branch-city: String, assets: real)

BankAccount(accno: int, branch-name: String, balance: real)

BankCustomer (customer-name: String, customer-street: String, customer-city: String)

Depositer(customer-name: String, accno: int)

LOAN (loan-number: int, branch-name: String, amount: real)

- i. Find all the customers who have an account at all the branches located in a specific city (Ex. Delhi).
- ii. Find all customers who have a loan at the bank but do not have an account.
- iii. Find all customers who have both an account and a loan at the Bangalore branch .
- iv. Find the names of all branches that have greater assets than all branches located in Bangalore.
- v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).
- vi. Update the Balance of all accounts by 5%

Queries :

- i. **Find all the customers who have an account at all the branches located in a specific city**

(Ex. Delhi). SELECT customer_name FROM BankCustomer WHERE customer_city = 'Delhi' AND NOT EXISTS (SELECT branch_name FROM Branch WHERE branch_city = 'Delhi' AND NOT EXISTS (SELECT * FROM BankAccount WHERE BankAccount.branch_name = Branch.branch_name AND BankCustomer.customer_name = Depositer.customer_name));

customername
Nikil
Ravi

- ii. **Find all customers who have a loan at the bank but do not have an account.**

SELECT customer_name FROM BankCustomer WHERE EXISTS (SELECT * FROM Loan WHERE Loan.branch_name = Branch.branch_name AND NOT EXISTS (SELECT * FROM BankAccount WHERE BankAccount.branch_name = Branch.branch_name AND BankCustomer.customer_name = Depositer.customer_name));

customername
Mohan

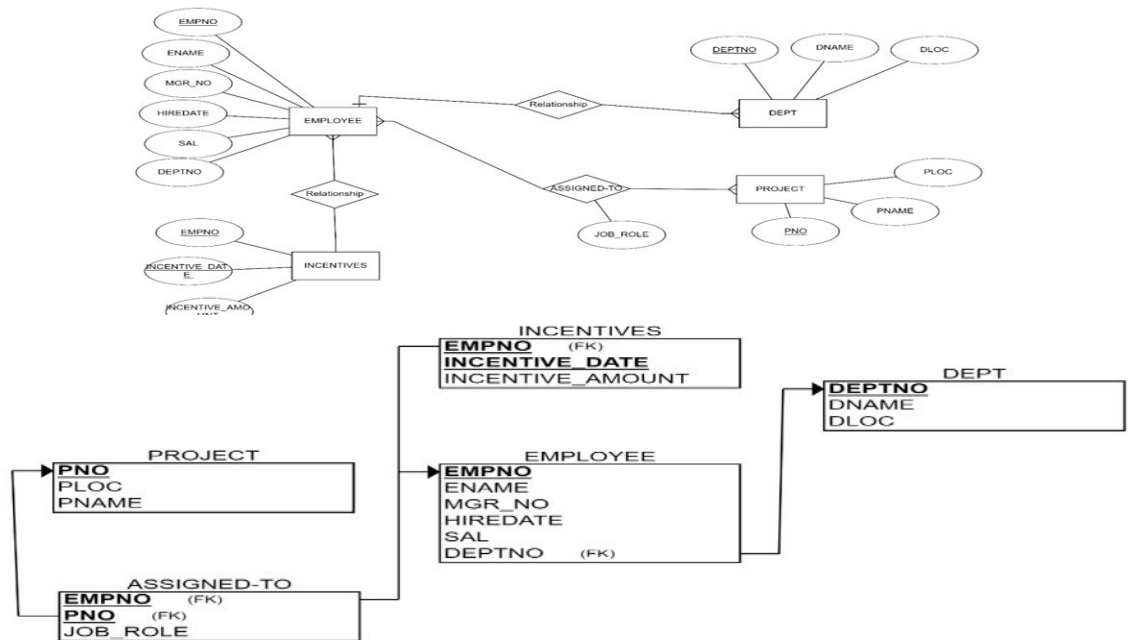
- iii. **Find all customers who have both an account and a loan at the Bangalore branch.**

SELECT DISTINCT customer_name FROM BankCustomer WHERE EXISTS (SELECT * FROM BankAccount WHERE BankAccount.branch_name = 'SBI_ResidencyRoad' AND BankCustomer.customer_name = Depositer.customer_name) AND EXISTS (SELECT * FROM Loan WHERE Loan.branch_name = 'SBI_ResidencyRoad' AND BankCustomer.customer_name = Depositer.customer_name);

customername
Avinash
Dinesh
Nikil
Ravi

5. Employee Database

PROGRAM 5: Employee Database



- Using Scheme diagram, create tables by properly specifying the primary keys and the foreign keys.
- Enter greater than five tuples for each table.
- Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru.
- Get Employee IDs of those employees who didn't receive incentives.
- Write a SQL query to find the employees name, number, dept, job_role, department location and project location who are working for a project location same as his/her department location.

Creating of database and tables:

```
create database employee_244;
```

```
use employee_244;
```

```
create table project(
pno int, ploc varchar(40),
pname varchar(40),
PRIMARY KEY(pno)
);
```


Inserting values into the tables:

```
insert into project values(1,"Bengaluru","Syntax");
insert into project values(2,"Gujurat","Rolex");
insert into project values(3,"Mysuru","Hybrid");
insert into project values(4,"Hyderabad","Synergy");
insert into project values(5,"Mumbai","Mercury");
select * from project;
```

pno	ploc	pname
1	Bengaluru	Syntax
2	Gujurat	Rolex
3	Mysuru	Hybrid
4	Hyderabad	Synergy
5	Mumbai	Mercury
NULL	NULL	NULL

```
insert into dept values(10,"Sales","Bengaluru");
insert into dept values(20,"Finance","West Bengal");
insert into dept values(30,"Marketing","Bihar");
insert into dept values(40,"Purchase","Mumbai");
insert into dept values(50,"Research & Development","Hyderabad");
select * from dept;
```

deptno	dname	dloc
10	Sales	Bengaluru
20	Finance	West Bengal
30	Marketing	Bihar
40	Purchase	Mumbai
50	Research & Development	Hyderabad
NULL	NULL	NULL

```
insert into employee values(100,"Prannay",400,'2003-01-01',100000,10);
insert into employee values(200,"Farhaan",500,'2004-02-02',100500,50);
insert into employee values(300,"Sanika",100,'2003-01-21',200500,30);
insert into employee values(400,"Sakshi", NULL ,'2008-02-17',300500,40);
insert into employee values(500,"Nishith",300,'2004-03-05',200700,40);
insert into employee values(600,"Sohan",200,'2005-11-01',200000,20);
insert into employee values(700,"Mahima",200,'2005-11-21',200900,20);
select * from employee;
```

empno	ename	mgr_no	hiredate	sal	deptno
100	Prannay	400	2003-01-01	100000	10
200	Farhaan	500	2004-02-02	100500	50
300	Sanika	100	2003-01-21	200500	30
400	Sakshi	NULL	2008-02-17	300500	40
500	Nishith	300	2004-03-05	200700	40
600	Sohan	200	2005-11-01	200000	20
700	Mahima	200	2005-11-21	200900	20
NULL	NULL	NULL	NULL	NULL	NULL

```
insert into incentives values(100,'2012-02-17',6000);
insert into incentives values(200,'2012-05-21',7000);
insert into incentives values(400,'2012-07-25',6500);
insert into incentives values(500,'2013-04-19',7400);
insert into incentives values(600,'2013-08-08',8000);
select * from incentives;
```

empno	incentive_date	incentive_amount
100	2012-02-17	6000
200	2012-05-21	7000
400	2012-07-25	6500
500	2013-04-19	7400
600	2013-08-08	8000
NULL	NULL	NULL

```
insert into assigned_to values(100,1, "Project Manager");
insert into assigned_to values(200,1, "Resource Manager");
insert into assigned_to values(300,2, "Business Analyst");
insert into assigned_to values(400,3, "Business Analyst");
insert into assigned_to values(500,3, "Project Manager");
insert into assigned_to values(600,5, "Resource Manager");
select * from assigned_to;
```

empno	pno	job_role
100	1	Project Manager
200	1	Resource Manager
300	2	Business Analyst
400	3	Business Analyst
500	3	Project Manager
600	5	Resource Manager

Queries:

- Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru.

```
select a.empno Employee_number from project p, assigned_to a
where p.pno=a.pno and p.ploc in("Hyderabad","Bengaluru","Mysuru");
```

Employee_number
100
200
400
500

- Get Employee ID's of those employees who didn't receive incentives.

```
select e.empno from employee e
where e.empno NOT IN (select i.empno from incentives i);
```

empno
700
300
NULL

6. More Queries on Employee Database

PROGRAM 6: More Queries on Employee Database

- i. Using Scheme diagram (under Program-5), Create tables by properly specifying the primary keys and the foreign keys.
- ii. Enter greater than five tuples for each table.
- iii. List the name of the managers with the maximum employees.
- iv. Display those managers name whose salary is more than average salary of his employee.
- v. Find the name of the second top level managers of each department.
- vi. Find the employee details who got second maximum incentive in January 2019.
Display those employees who are working in the same department where his manager is working.

Queries:

- iii. List the name of the managers with the maximum employees `select e1.ename`

```
from employee e1, employee e2 where e1.empno=e2.mgr_no group by e1.ename
having count(e1.mgr_no)=(select count(e1.ename) from employee e1, employee
e2 where e1.empno=e2.mgr_no group by e1.ename order by count(e1.ename) desc
limit 1);
```

ename
Farhaan

- iv. Display those managers name whose salary is more than average salary of his employee .
`select m.ename from employee m where m.empno in`
`(select mgr_no from employee)`
`and m.sal>(select avg(n.sal) from employee n`
`where n.mgr_no=m.empno);`

ename
Sakshi
Nishith

- v. Find the name of the second top level managers of each department.
`select ename from employee where empno in(select distinct mgr_no from`
`employee where empno in`
`(select distinct mgr_no from employee where empno in`
`(select distinct mgr_no from employee)));`

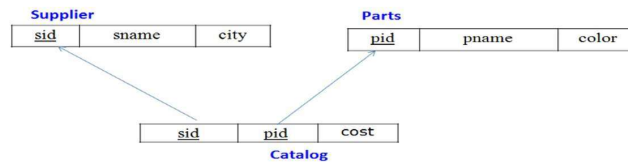
ename
Prannay
Sanika
Sakshi

- vi. Find the employee details who got second maximum incentive in January 2019.
`select * from employee where empno= (select i.empno from incentives i where`
`i.incentive_amount= (select max(n.incentive_amount) from incentives n where`
`n.incentive_amount<(select max(inc.incentive_amount) from incentives inc where`
`inc.incentive_date between '2019-01-01' and '2019-12-31') and incentive_date between '2019-`
`01-01' and '2019-12-31'));`

empno	ename	mgr_no	hiredate	sal	deptno
NULL	NULL	NULL	NULL	NULL	NULL

7. Supplier Database

PROGRAM 7: Supplier Database



- i. Using Scheme diagram, create tables by properly specifying the primary keys and the foreign keys.
- ii. Insert appropriate records in each table.
- iii. Find the pnames of parts for which there is some supplier. iv. Find the snames of suppliers who supply every part.
- v. Find the snames of suppliers who supply every red part.
- vi. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else. vii. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
- viii. For each part, find the sname of the supplier who charges the most for that part.

Creating database and table:

```
create database supplier_244;
use supplier_244;
```

```
create table Supplier
(
    sid int primary key,
    sname varchar(35),
    city varchar(35));
```

```
create table parts
(
    pid int primary key,
    pname varchar(35),
    color varchar(35));
```

```
create table catalog
(
    sid int,
    pid int,
    cost float,
    primary key(sid,pid),
    foreign key(sid) references
    Supplier(sid),
    foreign key(pid) referencesparts(pid));
```

Inserting values to tables:

iii. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).

create view c as select c.pid,p.pname,avg(cost) as co from catalog c,parts p where c.pid=p.pid group by c.pid; select ca.sid from catalog ca,c where ca.pid=c.pid and ca.cost>c.co and c.pid=ca.pid;

sid
10002
10004

iv. For each part, find the sname of the supplier who charges the most for that part.
select sname,co.pid,pname,cost from Supplier s,parts po,catalog co where co.pid=po.pid and s.sid=co.sid and co.cost = (select max(cost) from catalog where pid=po.pid) ;

sname	pid	pname	cost
Acme Widget	20001	Book	10
Acme Widget	20004	Mobile	10
Acme Widget	20005	Charger	10
Johns	20001	Book	10
Johns	20002	Pen	20
Reliance	20003	Pencil	40