2075-Baisakh

```
abindra old questi...-S0UPIIQ\ACER (72))*
create database Baishak2075;
use Baishak2075;
create table tblsalesman(
s_id int NOT NULL,
sname varchar(30),
city varchar(30),
commission int,
primary key (s_id)
);
create table tblcustomer(
c_id int NOT NULL,
cname varchar(30),
c_city varchar(30),
grade int,
s_id int foreign key references tblsalesman(s_id),
primary key (c_id)
);
create table tblorders(
ord_no int NOT NULL,
prch_amt int,
ord_date date,
c_id int foreign key references tblcustomer(c_id),
s_id int foreign key references tblsalesman(s_id),
primary key (ord_no)
);

insert into tblsalesman values (1,'Govind','Kathmandu',1200),(2,'Hari','Pokhara',2300),(3,'Jeevan','Humla',2000),(4,'Naira','Kathmandu',1000),
(5,'Gaurab','Birgunj',1800),(6,'Sapana','Lalitpur',1800),(7,'Gokarna','Humla',5000);
insert into tblcustomer values (1,'Nameera','Lalitpur',9,6),(2,'Amita','Kathmandu',10,1),(3,'Hari','Pokhara',11,2),(4,'Sona','Kathmandu',13,4),
(5,'Seema','Humla',6,3),(6,'Pushpa','Birgunj',12,4);
insert into tblorders values (1,3000,'2017-07-17',1,6),(2,5000,'2017-06-17',2,1),(3,7000,'2017-07-17',3,2),(4,3000,'2020-01-17',4,4),
(5,1000,'2020-01-19',5,3),(6,2000,'2020-01-19',6,5);
select *from tblsalesman;
select *from tblcustomer;
select *from tblorders;
```

DISCUSSION:

Here, we created a database 'Baishak2075' and created tables: 'tblsalesman', 'tblcustomer', 'tblorders' inside this database. We inserted various values into these tables with necessary primary key and foreign key constraints.

i. **Display the info of salesman whose name starts with n and has 4th character as r**

```
--display the info of salesman whose name starts with n and has 4th character as r
select *from tblsalesman where sname like 'N__r%';
```

DISCUSSION:

Here we selected and displayed the info of salesman from table 'tblsalesman' where the salesman's name begins with 'n' and its 4th character is 'r' by using SELECT command.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	s_id	sname	city	commission
1	4	Naira	Kathmandu	1000

ii. Display the count of customers with grades above Kathmandu's average

```
--display the count of customers with grades above kathmandu's average
select count(c_id) as no_of_customers from tblcustomer where grade>(select avg(grade) from tblcustomer where c_city='Kathmandu');
```

DISCUSSION:

Here, we used the concept of nested loop join. Firstly, we selected the average grade of the customer from Kathmandu city using aggregate function: avg(grade). Then, we selected the customer whose grade were greater than average and counted them using aggregate function: count(cid).

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	no_of_customers
1	2

iii. Increase the commission of salesmen by 2% if they are from Humla

```
--increase the commision of salesmen by 2% if they are from humla
update tblsalesman set commission = case
when city='Humla' then 1.02*commission
else 1*commission
end;
```

DISCUSSION:

Here, we used the command: Update and case. We created a condition: if the salesman is from Humla then increase its salary by 2%. Finally, SELECT command was used to display the result.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	s_id	sname	city	commission
1	1	Govind	Kathmandu	1200
2	2	Hari	Pokhara	2300
3	3	Jeevan	Humla	2040
4	4	Naira	Kathmandu	1000
5	5	Gaurab	Birgunj	1800
6	6	Sapana	Lalitpur	1800
7	7	Gokarna	Humla	5100

iv. Find the highest purchase amount on a date '2017-07-17' for each salesman with their ID.

```
--display each salesman with their id for highest purchase on '2017-07-17'
select s_id, max(prch_amt) from tblorders where ord_date='2017-07-17' group by s_id;
```

DISCUSSION:

Here, we displayed the highest purchase amount on a date '2017-07-17' for each salesman with their ID using function max() and group by.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	s_id	(No column name)
1	2	7000
2	6	3000

2074-Bhadra

```sql
create database Bhadra2074;
use Bhadra2074;
create table Employee(
empid int NOT NULL,
ename varchar(30),
age int,
salary bigint,
primary key (empid)
);
create table Department(
deptid int NOT NULL,
dname varchar(30),
budget bigint,
managerid int NOT NULL,
primary key(deptid),
);
create table works(
empid int foreign key references Employee(empid),
deptid int foreign key references Department(deptid),
hours int
);
insert into Employee values (1,'Shyam',36,60000),(2,'Gopal',55,67000),(3,'Hari',45,65000),(4,'Pushpa',33,40000),(5,'Anjali',39,46000);
insert into Employee values (6,'Manoj',47,60000);
insert into Department values (1,'BCT',400000,1),(2,'BCE',450000,2),(3,'BEL',350000,3),(4,'BEI',340000,4),(5,'Architecture',480000,5);
insert into works values (1,1,9),(2,2,8),(3,3,7),(4,4,4),(5,5,9);
select *from Employee;
select *from Department;
select *from works;
```

DISCUSSION:

Here, we created a database 'Bhadra2074' with tables: 'Employee', 'Department', 'works'. We inserted different values into these tables using insert command.

i. Display the name of department whose employee earns the maximum salary

```sql
--display the name of depart whose employee earns max salary
select dname,max(salary) as max_salary from Employee join works on Employee.empid=works.empid join Department on Department.deptid=works.deptid group by dname;
```

DISCUSSION:

Here, we joined the table employee with works and joined the resulting table with department. Finally, the department name and its maximum salary was displayed using SELECT command and aggregate function: max(salary).

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	dname	max_salary
1	Architecture	46000
2	BCE	67000
3	BCT	60000
4	BEI	40000
5	BEL	65000

ii. Display the name of employee, department, and number of hours they work

```
--display the name of employee,department and number of hours they work
select ename,dname,works.hours from Employee join works on Employee.empid=works.empid join Department on Department.deptid=works.deptid;
```

DISCUSSION:

Here, we joined the table Employee with works and the resulting table was joined with department. Finally, SELECT command was used to display the result.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	ename	dname	hours
1	Shyam	BCT	9
2	Gopal	BCE	8
3	Hari	BEL	7
4	Pushpa	BEI	4
5	Anjali	Architecture	9

iii. Give 20% raise in salary whose age is between 45 to 50 years

```
--give 20% raise in salary whose age is between 45 to 50 years
update Employee set salary=case
when age>45 and age<50 then 1.2*salary
else 1*salary
end;
```

DISCUSSION:

Here, we used the command: update and case. We set the condition of age greater than 45 and less than 50 to get an increment of 20%. Finally, SELECT command was used to display the output.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	empid	ename	age	salary
1	1	Shyam	36	60000
2	2	Gopal	55	67000
3	3	Hari	45	65000
4	4	Pushpa	33	40000
5	5	Anjali	39	46000
6	6	Manoj	47	72000

Bhadra-2073

```sql
create database bhadra2073;
use bhadra2073;
create table Employee(
ename varchar(30),
street varchar(30),
city varchar(30),
primary key (ename)
);
create table Company(
comp_name varchar(30) NOT NULL,
comp_city varchar(30),
primary key(comp_name)
);
create table Works(
ename varchar(30) foreign key references Employee(ename),
comp_name varchar(30) foreign key references Company(comp_name),
salary bigint
);
create table Manages(
ename varchar(30) foreign key references Employee(ename),
manager_name varchar(30)
);
insert into Employee values ('Gopal','kat','Ktm'),('Manish','pok','Pokhara'),('Swopnil','kat','Ktm'),('Ganesh','hed','Hetauda'),
                            ('Arpan','bhakt','Bhaktapur'),('Harry','lalit','Lalitpur');
insert into Company values ('NMB Bank','Ktm'),('NIC Bank','Lalitpur'),('Nabil Bank','Pokhara'),('SBI Bank','Hetauda');
insert into Works values ('Gopal','NMB Bank',60000),('Manish','NIC Bank',40000),('Swopnil','Nabil Bank',25000),('Harry','NIC Bank',47000),('Ganesh','SBI Bank',30000);
insert into Manages values ('Gopal','Manisha'),('Swopnil','Baburaj'),('Ganesh','Kane'),('Manish','Agab');
```

DISCUSSION:

Here, we created a database ‘bhadra2073’ and created tables: ‘employee’, ‘company’, ‘works’ and ‘manages’. We inserted different values into these tables.

i. **employee name, their company name where city end with pur**

```sql
--employee name, their company name where city end with pur
select Employee.ename,comp_name,city from Employee join Works on Employee.ename=Works.ename where Employee.city like '%pur';
```

DISCUSSION:

Here, we joined the table: Employee and works and selected the data with city that ends with ‘pur’. Finally, SELECT command was used to display the output.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	ename	comp_name	city
1	Harry	NIC Bank	Lalitpur

ii. increase the salary of employee by 25% whose salary is less than 30000

```
--increase the salary of employee by 25% whose salary is less than 30000
update Works set salary= case
when salary<30000 then 1.25*salary
else 1*salary
end;
```

DISCUSSION:

Here, we used the command: update and case. We set the condition of salary less than 30000 to get an increment of 25%. Finally, SELECT command was used to display the output.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	ename	comp_name	salary
1	Gopal	NMB Bank	60000
2	Manish	NIC Bank	40000
3	Swopnil	Nabil Bank	31250
4	Harry	NIC Bank	47000
5	Ganesh	SBI Bank	30000

Magh-2073

abindra q2.sql - D...-S0UPIIQ\ACER (60))* | abindra old questi...-S0UPIIQ\ACER (69))*

```sql
create database magh_2073;
use magh_2073;
create table employee(
empid int NOT NULL,
ename varchar(30),
eaddress varchar(30),
manager_id int,
primary key (empid)
);
create table department(
deptid int NOT NULL,
dname varchar(30),
primary key(deptid)
);
create table project(
pid int NOT NULL,
title varchar(30),
budget bigint,
deptid int foreign key references department(deptid),
primary key(pid)
);
create table works(
empid int foreign key references employee(empid),
pid int foreign key references project(pid),
hours int
);
insert into employee values (1,'Suresh','Kathmandu',1),(2,'Hari','Pokhara',2),(3,'Hari','Birgunj',3),(4,'Arush','Hetauda',4);
insert into department values (1,'BCE'),(2,'BCT'),(3,'BEL'),(4,'BEI');
insert into project values (1,'IOT',500000,4),(2,'Artificial Intelligence',79000,3),(3,'Medical',80000,2),(4,'Security',75000,1);
insert into works values (1,1,25),(2,2,17),(3,3,18),(4,4,16);
```

DISCUSSION:

Here, we created a database 'magh_2073' and created tables: 'employee', 'department', 'works' and 'project'. We inserted different values into these tables.

i. **display the employees who works on project with highest budget**

```sql
--display the employees who works on project with highest budget
select ename,max(budget) from employee join works on employee.empid=works.empid join project on project.pid=works.pid group by ename;
```

DISCUSSION:

Here, we joined the table employee with works and joined the resulting table with project. Finally, the employee name and its maximum budget was displayed using SELECT command and aggregate function: max(budget).

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	ename	(No column name)
1	Arush	75000
2	Hari	80000
3	Suresh	500000

ii. update the budget of all projects by 20% where budget less than 80000

```
update project
set budget = case
when budget<80000 then 1.2*budget
else 1*budget
end;
```

DISCUSSION:

Here, we used the command: update and case. We set the condition of budget= case to get a condition where budget is less than 80000 and increase it by 20%. Finally, SELECT command was used to display the output.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	pid	title	budget	deptid
1	1	IOT	500000	4
2	2	Artificial Intelligence	94800	3
3	3	Medical	80000	2
4	4	Security	90000	1

2072 Ashwin:

```
abindra old questi...-S0UPIIQ\ACER (68))*   abindra q2.sql - D...-S0UPIIQ\ACER (60))*
create database ashwin2072;
use ashwin2072;
drop table employee;
create table employee(
eno int NOT NULL,
ename varchar(30),
eaddress varchar(30),
edepart varchar(30),
phone bigint,
salary bigint
);
insert into employee values (1,'Abindra','Ktm','BCT',9845514896,40000),(2,'Aashsish','Pokhara','BCT',9845525678,50000),(3,'Hari','Hetauda','BCE',9845532243,65000),
(4,'Suman','Pokhara','BCE',9845545566,75000);
```

DISCUSSION:

Here, we created a database 'ashwin2072' and created table: 'employee'. We inserted different values into these tables.

i. **display the employee who earns more than 40k,works in bct and name contains a**

```
--display the employee who earns more than 40k,works in bct and name contains a
select *from employee where salary>40000 and edepart='BCT' and ename like '%a%';
```

DISCUSSION:

Here, we selected the info of employee whose salary is greater than 40000, who belongs to 'BCT' department and whose name contains alphabet 'a'.

RELATIONAL ALGEBRA:

OUTPUT:

Results | Messages

	eno	ename	eaddress	edepart	phone	salary
1	2	Aashsish	Pokhara	BCT	9845525678	50000

ii. **increase the salary of employee who earns less than 50000**

```
--increase the salary of employee who earns less than 50000
update employee set salary=case
when salary<50000 then 1.25*salary
else 1*salary
end;
```

DSICUSSION:

Here, we used the command: update and case. We set the condition of salary less than 50000 to get an increment of 25%. Finally, SELECT command was used to display the output.

OUTPUT:

Results | Messages

	eno	ename	eaddress	edepart	phone	salary
1	1	Abindra	Ktm	BCT	9845514896	50000
2	2	Aashsish	Pokhara	BCT	9845525678	50000
3	3	Hari	Hetauda	BCE	9845532243	65000
4	4	Suman	Pokhara	BCE	9845545566	75000