

Experiment no: 02Date: +Aim: Write a query to demonstrate logical operators.Logical Operators: There are 3 types of logical operators

- (i) AND: Join two (or) more conditions and return result only when all the conditions are true.
- (ii) OR: Join two (or) more conditions and return result only when any condition is true.
- (iii) NOT: Negation.

Aim: 1 - Write a query to display the records from table student who are belongs to Tenali with name Srinivas.Student:

SNO	SNAME	SADDRESS	TOTAL
1	Srinivas	Tenali	900
2	Krishna	Guntur	700
3	Vasu		
4	Raju	tenali	700

Query: select * from STUDENT where
SADDRESS = 'tenali' AND SNAME = 'Srinivas';

Output:

SNO	SNAME	SBRANCH	SADDRESS
1	Srinivas	Tenali	900

Aim: 2 - Write a query to display records from table student who belongs to either Tenali or Guntur.

Experiment no: 02Date: +Aim: Write a query to demonstrate logical operators.Logical Operators: There are 3 types of logical operators

- (i) AND: Join two (or) more conditions and return result only when all the conditions are true.
- (ii) OR: Join two (or) more conditions and return result only when any condition is true.
- (iii) NOT: Negation.

Aim: 1 - Write a query to display the records from table student who belongs to Tenali with name Srinivas.Student:

SNO	SNAME	SADDRESS	TOTAL
1	Srinivas	Tenali	900
2	Krishna	Guntur	700
3	Vasu		
4	Raju	tenali	700

Query: select * from STUDENT where
 SADDRESS = 'tenali' AND SNAME = 'Srinivas';

Output:

SNO	SNAME	SBRANCH	SADDRESS
1	Srinivas	Tenali	900

Aim 2: Write a query to display records from table student who belongs to either Tenali or Guntur.

Query:- select ~~SN~~ from student where SADDRESS
NOT IN ('tenali', 'guntur');

Output:-

3

(ii) BETWEEN (or) RANGES:- It returns true, if columnname is existed in the range of columnvalue1 and columnvalue2.

Syntax:- Columnname BETWEEN columnvalue1 AND
columnvalue2;

Aim:- Write a query to display SNAME where total ranges between 900 & 700.

Query:- select SNAME from student where
TOTAL between 700 and 900;

Output:-

SNAME

Srinivas

Krishna

Raju.

(iii) LIKE (or) WILD CARDS:- Again like consists of 2 special characters.

1) percentage (%):- substituted by any expression (more than one character)

2) underscore (_):- substituted by single character.

Aim:- Write a query about students whose names are starting with 'K'.



V) EXISTS :- This operator returns true, a subquery returns atleast one row otherwise it is false.

Aim :- Write a query to demonstrate exists operator.

Query :- select * from student where exists
(select * from EMPLOYEE);

Output :-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	guntur	700
3	Vasu		
4	Raju	Tenali	700

Experiment no: 03

Date:

Aim1: Write a query to add the new record into the table EMPLOY

Insert: to add the new records into any table, we use this insert command.

Syntax: insert into tablename values
(attribute1value, attribute2value, ...);

Query: insert into EMPLOY values

(1, 'Srinivas', 'HA', '10-feb-2023', 25000);

Output: 1 row created.

Aim2: Write a query to add new record into the table

DEPART

Query: insert into DEPART values (10, 'accounting', 'NewYork');

Output: 1 row created.

Aim3: Write a query to list all the records from the table EMPLOY.

Select: It is used to display all the records from the table.

Syntax: select * from tablename;

Query: select * from EMPLOY;

Output: 1 Srinivas HA 10-feb-2023 25000.

Aim4: Write a query to display the records from the table DEPART.

Query: select * from DEPART;

Output: 10 accounting NewYork.

Syntax:- Delete Tablename;

Query:- Delete DEPART;

Output:- table deleted.

Aim:- Write a query to delete records from table EMPLOY where job is clerk.

Delete command using where clause:- We can also use the delete command for deleting particular work (or) records by using the where clause.

Syntax:- Delete tablename where condition;

Query:- Delete EMPLOY where job = 'clerk';

Output:- table deleted.

Aim:- Write a query to modify salary column in table EMPLOY, for each EMPLOY salary is 20000.

Update:- It is used to change table data. Using update command with where clause, to change a particular record in database.

Syntax:- Update tablename set column1=value1,
column2=value2, --- where condition;

Query:- Update EMPLOY set salary=20000;

Output:- 2 rows updated.

Aim:- Write a query to modify the salary column in the table EMPLOY to add 2000/- for each employ.

Output :-

<u>ENO</u>	<u>ENAME</u>	<u>ETOB</u>	<u>ESALARY</u>	<u>EDEPTNO</u>
5	Ramu	1000	clerk	20
6	Suresh	1000	clerk	30
4	Raju	2000	clerk	10
7	Feld	3000	Analyst	10
3	Vasu	4000	manager	30
1	Srinivas	5000	manager	10
2	Jones	5000	manager	20

Aim:- Write a query to retrieve information about the employees with respect to salaries in descending order.

Query:- select * from EMPLOY ORDER BY SALARY DESC;

Output:-

<u>ENO</u>	<u>ENAME</u>	<u>ETOB</u>	<u>ESALARY</u>	<u>EDEPTNO</u>
1	Srinivas	manager	5000	10
2	Jones	manager	5000	20
3	Vasu	manager	4000	30
7	Feld	analyst	3000	10
4	Raju	clerk	2000	10
6	Suresh	clerk	2000	30
5	Ramu	clerk	1000	20

Table TEST:-

<u>ENO</u>	<u>ENAME</u>	<u>ESAL</u>	<u>ECOMM</u>	<u>EDEPTNO</u>
1	Srinivas	5000	10	10
2	Jones	5000	10	20
3	Vasu	4000	10	30

4	Raju	2000	10	10
5	Ramu	1000	10	20
6	Suresh	1000	10	30
7	Fold	3000		10
8	Blake	2000	20	20

Aim 3: Write a query to calculate the average salary of the employees in each department in the table TEST.

Query 1: select avg(sal) from TEST where DEPTNO=10;

Query 2: select avg(sal) from TEST where DEPTNO=20;

Query 3: select avg(sal) from TEST where DEPTNO=30;

O/P:

DEPT NO	AVG(SAL)
10	3333.33
20	2666.66
30	2500

We can get the same result by single line using group by clause.

Query: select DEPTNO, avg(sal) from TEST Group by DEPTNO;

Output:

DEPT NO	AVG(SAL)
10	3333.33
20	2666.66
30	2500

Aim 4: Write a query if you want retrieve DEPT's contained more than 2 employees in table TEST.

Query: select DEPTNO, COUNT(DEPTNO) from TEST group by DEPTNO HAVING COUNT (DEPTNO) > 2;

O/P:

DEPTNO	COUNT(DEPTNO)
10	3
20	3

Experiment no: 4

Aim: Queries using order by clause, group by clause and having clause.

Date:

Description:

Order by clause: The order by clause is used for rows have displayed in sorting order. If used, the 'order by' must be the last clause in 'select' statement.

Syntax: order by columnname [ASC/DESC];

here, columnname refers to the field.

ASC - ascending order

DESC - descending order.

Group by clause: It is used to arrange the data in a table to form the certain groups.

Syntax: group by columnname1, columnname2, ---;

Having clause: It selects the specific groups. The having clause compare some property of group with a constant value, if a group satisfied the condition in the having clause, it is included query result.

Aim: Write a query to retrieve information about the employees with respect to salaries in ascending order.

Query: select * from EMPLOY order by SALARY ASC;

Aims:- Write a query to display the employ job, name from the table EMPLOY.

Query:- select ENAME, JOB from EMPLOY;

Output:-

ENAME	EJOB
Srinivas	HA

Aims:- Write a query to display all EMPLOY NAMES from the table EMPLOY whose job is manager.

Select:- It is also used to display the records with particular work by using the where clause

Syntax:- select * from tablename where EJOB = 'Manager';

Query:- select ENAME, SALARY from EMPLOY
where job = 'manager';

Output:-

ENAME	SAL
Srinivas	25000

Aims:- Write a query to list all the database tables available in the current user.

Query:- select * from TAB;

Output:-

TNAME	TABTYPE
EMPLOY	table
STUDENT	table
DEPART	table.

Aims:- Write a query to delete all records from the table DEPART.

Delete:- It is used to delete all the records from the table.

Aim1:- Write a query to display records from table student whose total is not equal to 700.

Query:- select * from student where TOTAL != 700;

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
3	Vasu		

Aim:- Write a query to demonstrate special Operators.

Special Operators :- There are 5 operators in it,

(1) IN :- It returns true if the column name is one of the values in column values list.

Syntax:- Columnname IN (column values list);

Aim1:- Write a query to display the records from table student who belongs to either Tenali, guntur, Vijayawada, Hyderabad.

Query:- select * from STUDENT where SADDRESS IN('tenali', 'guntur', 'Vijayawada', 'Hyderabad');

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	guntur	700

Aim2:- Write a query to display SNO from table student who are not belongs either tenali or guntur.

Query select * from student where SNAME like 'K';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700

Aims:- Write a query about students whose names are starting with K and ending with na.

Query:- select * from STUDENT where SNAME like 'K%.na';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700

Aims:- Write a query about student names whose second character must be 's'.

Query:- select SNAME from STUDENT where SNAME like '-s%';

Output:- No rows selected.

(iv) ISNULL:- It is used to check for null attribute (column values).

Query:-

Select * from STUDENT where SADDRESS is null;

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
3	Vasu		

Query: select * from STUDENT where

SADDRESS = 'tenali' OR SADDRESS = 'guntur';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	Guntur	700

Aim:- Write a query to display records from table student who are belongs to either tenali or guntur and with name Srinivas.

Query: select * from STUDENT where SNAME = 'Srinivas'
AND (SADDRESS = 'tenali' OR SADDRESS = 'guntur');

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	guntur	700

Aim:- Write a query to display records from table student whose address is not Tenali.

Query:- select * from STUDENT where
not SADDRESS = 'tenali';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700
3	Vasu		

Aim - Write a query to demonstrate relational operators

Relational Operators - There are 6 types in it,

- (i) $<$ - less than
- (ii) $>$ - greater than
- (iii) $<=$ - less than or equal to
- (iv) $>=$ - greater than or equal to
- (v) $=$ - equals to
- (vi) $!=$ (or) $<>$ - not equal to.

Aim1 - Write a query to display records from table student who are getting total > 700 .

Query - select * from student where total > 700 ;

Output -

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900

Aim2 - Write a query to display records from table student who are getting total < 700

Query - select * from student where total < 700 ;

Output - No records selected.

Aim3 - Write a query to display records from table student whose address belongs to guntur.

Query - select * from student where SADDRESS='guntur';

Output -

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700

for adding 2000/- for each employ we use one arithmetic operator '+' to the current salary

Query:- Update EMPLOY set salary = salary + 2000;

Output:- 2 rows updated.

Alm12:- Write a query to modify the job column by manager in the table EMPLOY whose present job is clerk.

Query:- Update EMPLOY

set job = 'manager'
where job = 'clerk';

Output:-

table updated.

Reference table:- EMPLOY

<u>ENO</u>	<u>ENAME</u>	<u>EJOB</u>	<u>ESALARY</u>	<u>EXCEPTNO</u>
1	Srinivas	manager	5000	10
2	Jones	manager	5000	20
3	Vasu	manager	4000	30
4	Raju	clerk	2000	10
5	Ramu	clerk	1000	20
6	Suresh	clerk	1000	30
7	Pold	Analyst	3000	10

Experiment no: 02Date: +Aim: Write a query to demonstrate logical operators.Logical Operators: There are 3 types of logical operators

- (i) AND: Join two (or) more conditions and return result only when all the conditions are true.
- (ii) OR: Join two (or) more conditions and return result only when any condition is true.
- (iii) NOT: Negation.

Aim: 1 - Write a query to display the records from table student who are belongs to Tenali with name Srinivas.Student:

SNO	SNAME	SADDRESS	TOTAL
1	Srinivas	Tenali	900
2	Krishna	Guntur	700
3	Vasu		
4	Raju	tenali	700

Query: select * from STUDENT where
 SADDRESS = 'tenali' AND SNAME = 'Srinivas';

Output:

SNO	SNAME	SBRANCH	SADDRESS
1	Srinivas	Tenali	900

Aim 2: Write a query to display records from table student who belongs to either Tenali or Guntur.

Query:- select ~~SN~~ from student where SADDRESS
NOT IN ('tenali', 'guntur');

Output:-

3

(ii) BETWEEN (or) RANGES:- It returns true, if columnname is existed in the range of columnvalue1 and columnvalue2.

Syntax:- Columnname BETWEEN columnvalue1 AND
columnvalue2;

Aim:- Write a query to display SNAME where total ranges between 900 & 700.

Query:- select SNAME from student where
TOTAL between 700 and 900;

Output:-

SNAME

Srinivas

Krishna

Raju.

(iii) LIKE (or) WILD CARDS:- Again like consists of 2 special characters.

1) percentage (%):- substituted by any expression (more than one character)

2) underscore (_):- substituted by single character.

Aim:- Write a query about students whose names are starting with 'K'.



V) EXISTS :- This operator returns true, a subquery returns atleast one row otherwise it is false.

Aim :- Write a query to demonstrate exists operator.

Query :- select * from student where exists
(select * from EMPLOYEE);

Output :-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	guntur	700
3	Vasu		
4	Raju	Tenali	700

Experiment no: 03Date:

Aim1:- Write a query to add the new record into the table EMPLOY

Insert:- to add the new records into any table, we use this insert command.

Syntax:- insert into tablename values
(attribute1value, attribute2value, --);

Query:- insert into EMPLOY values

(1, 'Srinivas', 'HA', '10-feb-2023', 25000);

Output:- 1 row created.

Aim2:- Write a query to add new record into the table DEPART

Query:- insert into DEPART values (10, 'accounting', 'NewYork');

Output:- 1 row created.

Aim3:- Write a query to list all the records from the table EMPLOY.

Select:- It is used to display all the records from the table.

Syntax:- select * from tablename;

Query:- select * from EMPLOY;

Output:- 1 Srinivas HA 10-feb-2023 25000.

Aim4:- Write a query to display the records from the table DEPART.

Query:- select * from DEPART;

Output:- 10 accounting NewYork.

Syntax:- Delete Tablename;

Query:- Delete DEPART;

Output:- table deleted.

Aim:- Write a query to delete records from table EMPLOY where job is clerk.

Delete command using where clause:- We can also use the delete command for deleting particular work (or) records by using the where clause.

Syntax:- Delete tablename where condition;

Query:- Delete EMPLOY where job = 'clerk';

Output:- table deleted.

Aim:- Write a query to modify salary column in table EMPLOY, for each EMPLOY salary is 20000.

Update:- It is used to change table data. Using update command with where clause, to change a particular record in database.

Syntax:- Update tablename set column1=value1,
column2=value2, --- where condition;

Query:- Update EMPLOY set salary=20000;

Output:- 2 rows updated.

Aim:- Write a query to modify the salary column in the table EMPLOY to add 2000/- for each employ.

Output :-

<u>ENO</u>	<u>ENAME</u>	<u>ETOB</u>	<u>ESALARY</u>	<u>EDEPTNO</u>
5	Ramu	1000	clerk	20
6	Suresh	1000	clerk	30
4	Raju	2000	clerk	10
7	Fold	3000	Analyst	10
3	Vasu	4000	manager	30
1	Srinivas	5000	manager	10
2	Jones	5000	manager	20

Aim:- Write a query to retrieve information about the employees with respect to salaries in descending order.

Query:- select * from EMPLOY ORDER BY SALARY DESC;

Output:-

<u>ENO</u>	<u>ENAME</u>	<u>ETOB</u>	<u>ESALARY</u>	<u>EDEPTNO</u>
1	Srinivas	manager	5000	10
2	Jones	manager	5000	20
3	Vasu	manager	4000	30
7	Fold	analyst	3000	10
4	Raju	clerk	2000	10
6	Suresh	clerk	2000	30
5	Ramu	clerk	1000	20

Table TEST:-

<u>ENO</u>	<u>ENAME</u>	<u>ESAL</u>	<u>ECOMM</u>	<u>EDEPTNO</u>
1	Srinivas	5000	10	10
2	Jones	5000	10	20
3	Vasu	4000	10	30

4	Raju	2000	10	10
5	Ramu	1000	10	20
6	Suresh	1000	10	30
7	Fold	3000		10
8	Blake	2000	20	20

Aim 3: Write a query to calculate the average salary of the employees in each department in the table TEST.

Query 1: select avg(sal) from TEST where DEPTNO=10;

Query 2: select avg(sal) from TEST where DEPTNO=20;

Query 3: select avg(sal) from TEST where DEPTNO=30;

O/P:

DEPT NO	AVG(SAL)
10	3333.33
20	2666.66
30	2500

We can get the same result by single line using group by clause.

Query: select DEPTNO, avg(sal) from TEST Group by DEPTNO;

Output:

DEPT NO	AVG(SAL)
10	3333.33
20	2666.66
30	2500

Aim 4: Write a query if you want retrieve DEPT's contained more than 2 employees in table TEST.

Query: select DEPTNO, COUNT(DEPTNO) from TEST group by DEPTNO HAVING COUNT (DEPTNO) > 2;

O/P:

DEPTNO	COUNT(DEPTNO)
10	3
20	3

Experiment no: 4

Aim: Queries using order by clause, group by clause and having clause.

Date:

Description:

Order by clause: The order by clause is used for rows have displayed in sorting order. If used, the 'order by' must be the last clause in 'select' statement.

Syntax: order by columnname [ASC/DESC];

here, columnname refers to the field.

ASC - ascending order

DESC - descending order.

Group by clause: It is used to arrange the data in a table to form the certain groups.

Syntax: group by columnname1, columnname2, ---;

Having clause: It selects the specific groups. The having clause compare some property of group with a constant value, if a group satisfied the condition in the having clause, it is included query result.

Aim: Write a query to retrieve information about the employees with respect to salaries in ascending order.

Query: select * from EMPLOY order by SALARY ASC;

Aims:- Write a query to display the employ job, name from the table EMPLOY.

Query:- select ENAME, JOB from EMPLOY;

Output:-

<u>ENAME</u>	<u>EJOB</u>
Srinivas	HA

Aims:- Write a query to display all EMPLOY NAMES from the table EMPLOY whose job is manager.

Select:- It is also used to display the records with particular work by using the where clause

Syntax:- select * from tablename where EJOB = 'Manager';

Query:- select ENAME, SALARY from EMPLOY
where job = 'manager';

Output:-

<u>ENAME</u>	<u>SAL</u>
Srinivas	25000

Aims:- Write a query to list all the database tables available in the current user.

Query:- select * from TAB;

Output:-

<u>TNAME</u>	<u>TABTYPE</u>
EMPLOY	table
STUDENT	table
DEPART	table.

Aims:- Write a query to delete all records from the table DEPART.

Delete:- It is used to delete all the records from the table.

Aim1:- Write a query to display records from table student whose total is not equal to 700.

Query:- select * from student where TOTAL != 700;

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
3	Vasu		

Aim:- Write a query to demonstrate special Operators.

Special Operators :- There are 5 operators in it,

(1) IN :- It returns true if the column name is one of the values in column values list.

Syntax:- Columnname IN (column values list);

Aim1:- Write a query to display the records from table student who belongs to either Tenali, guntur, Vijayawada, Hyderabad.

Query:- select * from STUDENT where SADDRESS IN('tenali', 'guntur', 'Vijayawada', 'Hyderabad');

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	guntur	700

Aim2:- Write a query to display SNO from table student who are not belongs either tenali or guntur.

Query select * from student where SNAME like 'K';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700

Aims:- Write a query about students whose names are starting with K and ending with na.

Query:- select * from STUDENT where SNAME like 'K%.na';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700

Aims:- Write a query about student names whose second character must be 's'.

Query:- select SNAME from STUDENT where SNAME like '-s%';

Output:- No rows selected.

(iv) ISNULL:- It is used to check for null attribute (column values).

Query:-

Select * from STUDENT where SADDRESS is null;

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
3	Vasu		

Query: select * from STUDENT where

SADDRESS = 'tenali' OR SADDRESS = 'guntur';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	Guntur	700

Aim:- Write a query to display records from table student who are belongs to either tenali or guntur and with name Srinivas.

Query: select * from STUDENT where SNAME = 'Srinivas'
AND (SADDRESS = 'tenali' OR SADDRESS = 'guntur');

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900
2	Krishna	guntur	700

Aim:- Write a query to display records from table student whose address is not Tenali.

Query:- select * from STUDENT where
not SADDRESS = 'tenali';

Output:-

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700
3	Vasu		

Aim - Write a query to demonstrate relational operators

Relational Operators - There are 6 types in it,

- (i) $<$ - less than
- (ii) $>$ - greater than
- (iii) $<=$ - less than or equal to
- (iv) $>=$ - greater than or equal to
- (v) $=$ - equals to
- (vi) $!=$ (or) $<>$ - not equal to.

Aim1 - Write a query to display records from table student who are getting total > 700 .

Query - select * from student where total > 700 ;

Output -

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
1	Srinivas	Tenali	900

Aim2 - Write a query to display records from table student who are getting total < 700

Query - select * from student where total < 700 ;

Output - No records selected.

Aim3 - Write a query to display records from table student whose address belongs to guntur.

Query - select * from student where SADDRESS='guntur';

Output -

<u>SNO</u>	<u>SNAME</u>	<u>SADDRESS</u>	<u>TOTAL</u>
2	Krishna	Guntur	700

for adding 2000/- for each employ we use one arithmetic operator '+' to the current salary

Query:- Update EMPLOY set salary = salary + 2000;

Output:- 2 rows updated.

Alm12:- Write a query to modify the job column by manager in the table EMPLOY whose present job is clerk.

Query:- Update EMPLOY

set job = 'manager'
where job = 'clerk';

Output:-

table updated.

Reference table:- EMPLOY

<u>ENO</u>	<u>ENAME</u>	<u>EJOB</u>	<u>ESALARY</u>	<u>EDEPTNO</u>
1	Srinivas	manager	5000	10
2	Jones	manager	5000	20
3	Vasu	manager	4000	30
4	Raju	clerk	2000	10
5	Ramu	clerk	1000	20
6	Suresh	clerk	1000	30
7	Pold	Analyst	3000	10