

DATABASE SYSTEM (CS-329)

SUBMITTED BY
MUNIB-UL-HASSAN

ROLL NO #
CS19-037



SUBMITTED TO
SIR ASIF RAZA

DEPARTMENT OF COMPUTER SCIENCE
SIR SYED UNIVERSITY OF ENGINEERING AND TECHNOLOGY

INDEX:



LAB # 01

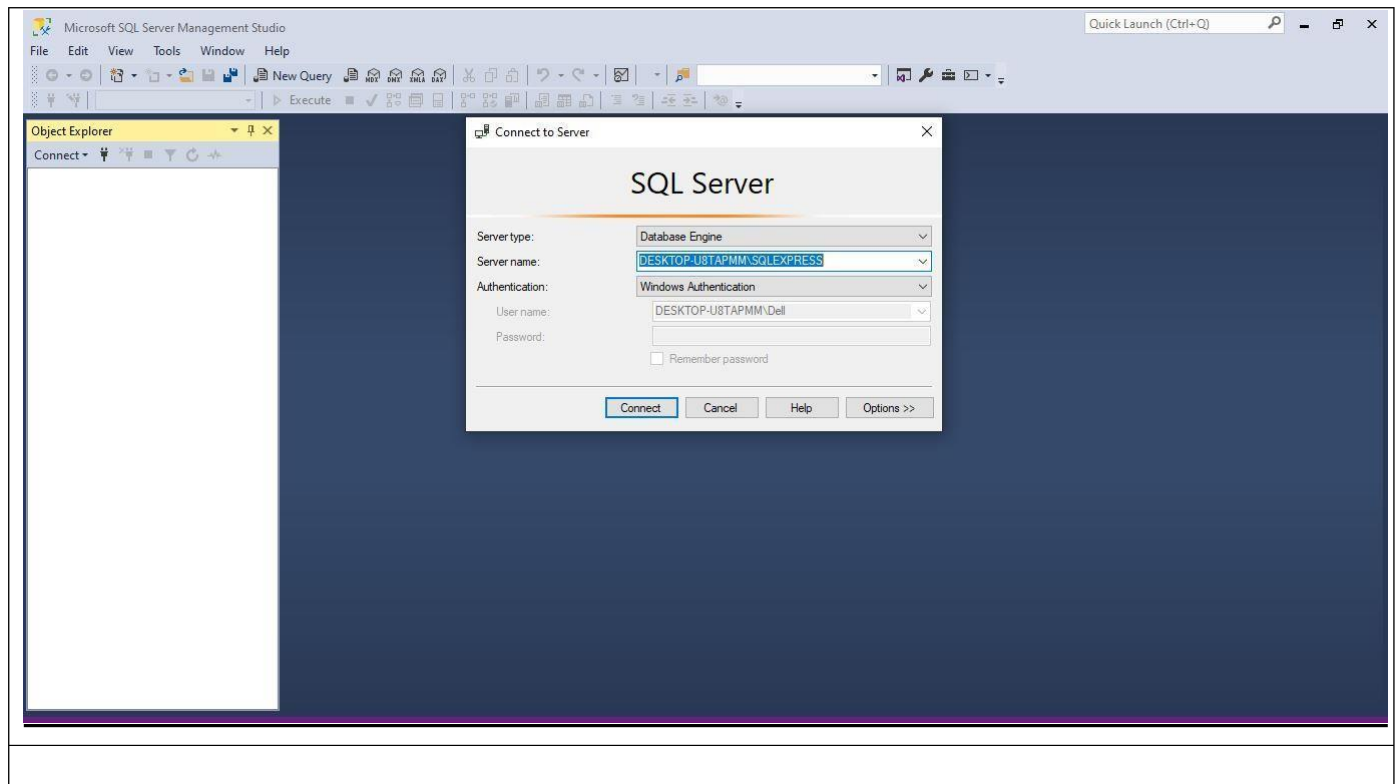
Introduction to Database and Microsoft SQL Server

TASK:

Visit the below link for the Installation of Microsoft SQL Server 2008.

<https://www.youtube.com/watch?v=4WEFTQ3VJNg>

<https://www.youtube.com/watch?v=WKWZZcrin5I>



LAB # 02

TASKS:

- Create a new table Person and insert at least 5 records.

The screenshot shows a SQL query window with the following code:

```
create table Persons (ID_no int , F_Name varchar(10), age int, Phone_No varchar(20),  
addres varchar(20));  
  
select * from Persons;  
  
insert into Persons values ( 05 , 'Ahmed' , 24 , 03458609458 , 'Bufferzone' );  
  
drop table Persons;
```

Below the query window, the 'Results' tab displays the data inserted into the 'Persons' table:

	serial_no	F_Name	age	Phone_No	addres
1	1	Ali	20	3158706847	North
2	2	Hamza	21	3158806047	Nazimabad
3	3	Osama	22	3138809548	PECHS
4	4	Daniyal	23	3438809458	DHA
5	5	Ahmed	24	3458609458	Bufferzone

- Create a new table Customer and insert at least 5 records.

The screenshot shows a SQL query window with the following code:

```
create table Customers (ID_no int , F_Name varchar(10), Email varchar(MAX), Phone_No varchar(20),  
addres varchar(20));  
  
select * from Customers;  
  
insert into Customers values ( 05 , 'Waheed' , 'Waheed98@gmail.com' , 03218537454 , 'Zamzama' );
```

Below the query window, the 'Results' tab displays the data inserted into the 'Customers' table:

	ID_no	F_Name	Email	Phone_No	addres
1	1	Hassan	Hassan123@gmail.com	3458609458	Bufferzone
2	2	Faizan	Faizan99@gmail.com	3158579458	North
3	3	Areeb	Areeb65@gmail.com	3358567454	DHA
4	4	Sanjay	Sanjay01@gmail.com	3238567454	SMCHS
5	5	Waheed	Waheed98@gmail.com	3218537454	Zamzama

- Create a new table Order and insert at least 5 records

SQLQuery1.sql - D:\U8TAPMM\ Dell (56)) * -p X

```

create table Orders (Order_No int , Cust_Name varchar(10), Email varchar(MAX), Phone_No varchar(20),
address varchar(20));

select * from Orders;

insert into Orders values ( 67534 , 'Haris' , 'Haris42@gmail.com' , 03435652732 , 'Johar' );

```

100 %

Results Messages

	Order_No	Cust_Name	Email	Phone_No	address
1	38335	Shazaib	Shazaib01@gmail.com	3418539354	Saddar
2	93798	Sahir	Sahir08@gmail.com	3157867262	FB AREA
3	93837	Raza	Raza24@gmail.com	345768765	Shadman
4	56782	Hamza	Hamza33@gmail.com	3476523243	Gulshan
5	67534	Haris	Haris42@gmail.com	3435652732	Johar

- Add a new field in the Person table.

```

Alter table Persons add weight_KG int;
Alter table Persons alter column weight_KG int;

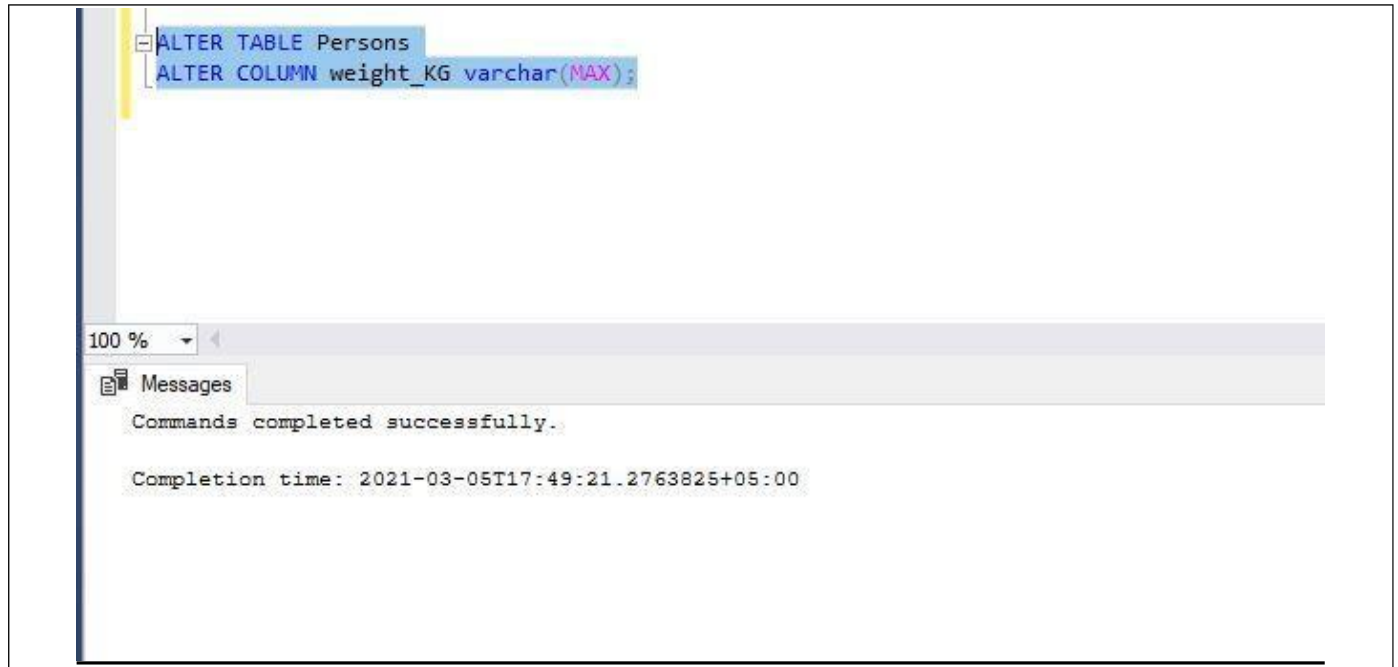
```

100 %

Results Messages

	serial_no	F_Name	age	Phone_No	address	weight_KG
1	1	Ali	20	3158706847	North	NULL
2	2	Hamza	21	3158806047	Nazimabad	NULL
3	3	Osama	22	3138809548	PECHS	NULL
4	4	Daniyal	23	3438809458	DHA	NULL
5	5	Ahmed	24	3458609458	Bufferzone	NULL

- **Modify any field in the Person table.**



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the command: `ALTER TABLE Persons
ALTER COLUMN weight_KG varchar(MAX);`. The bottom pane shows a message: `Commands completed successfully.` and the completion time: `2021-03-05T17:49:21.2763825+05:00`.

- **Use some constraints in your queries.**



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the command: `create table Orders
(Order_No int ,
Cust_Name varchar(10),
Email varchar(MAX),
Phone_No varchar(MAX),
addres varchar(MAX));`. The bottom pane shows the command: `select * from Persons;`

LAB # 03 & 04

TASK 1:

1. Insert new employee's records.
2. Insert new department records deptno=50, dname=ADVERTISING and loc=MIAMI.
3. Delete the records of Sales department
4. Change the employees sal for Smith
5. Confirm all these changes by using Select statement
6. Update the salary of each employee to 5000.
7. Change the salary to \$1,000 for all employees with a salary less than \$900.
8. Change the commission of department 20 to 1000.
9. Change the hire date of all clerks to 02-04-2000
10. Delete all the records having hire date before 21-dec-81
11. Delete all records where salary is greater than 2000.

TASK2:

1. Use insert, update & delete commands in 10 different scenarios.

ASSIGNMENTS:

1. List all rows of the table emp.
2. List all rows of the dept.
3. List all employees' number from emp.
4. List all employees name from emp.
5. List all departments' number from emp.
6. Find all employees whose salaries are between 500 and 1500.
7. Find all employees whose salaries are between 1500 and 2500.
8. Find all employees whose salaries are between 2600 and 5000.
9. Find all employees whose salaries are less than 2000.
10. Find all employees whose salaries are greater than 2000.
11. Find those employees whose mgr. are 7902, 7566, 7788.
12. Find those employees whose mgr. is between 7788 and 7092.
13. List all employees whose name starts with 's'.
14. List all employees whose name start with 'a'.
15. List all employees having first name as 'Scott'.
16. List all employees having first name as 'smith' or 'king'.

17. List all employees having first letter in their name 'w' or 'k' and they don't belong to Department number 10,20 and 40.
 18. List all employees whose name begin with 'Scott' and end with martin.
 19. List all employees whose name start with 'm' and belong to department number 30.
 20. List all employees whose name started with character 'b' and their jobs are manager.
 21. List all employees who do not have job manager.
 22. Find those employees whose job do not start with 'a'.
 23. Find those employees whose job does not start with 'c'.
 24. List of those employees whose mgr. is not null.
 25. List of those employees whose jobs are 'manager or clerk' of department number 10.
 26. List of those employees whose jobs are 'analyst' and 'salesman' of department 30.
 27. Find all clerks who earn salaries between 1000 and 2000.
 28. Find all managers who earn salaries between 2500 and 3000.
 29. Find all employees who are either clerk or 'manager' and all employees who earn Salaries in the range of 1000 and 2000.
 30. Find all employees who are either manager and/or all employees who earn salaries between 2000 and 3000.
 31. Find all employees whose salaries are equal to 1500 and jobs are manager or salesmen.
-

```
SQLQuery1.sql - D:\U8TAPMM\Del1 (62)*
create table Employee (Eid int primary key not null identity (1,10), EName varchar(Max), EHireDate varchar(MAX), ESalary int, ECommission int);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Sakhawat', 2016, 50000, 500);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Hamza', 2017, 50000, 500);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Osama', 2018, 40000, 600);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Raza', 2019, 30000, 700);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Haris', 2020, 20000, 800);
select * from Employee;
update Employee set ESalary = 10000;
select * from Employee;
update Employee set EHireDate = 2021 where EName = 'Hamza';
update Employee set ESalary = 15000 where EName = 'Osama';
update Employee set ESalary = 20000 where EName = 'Raza';
select * from Employee;
```

100 %

Results Messages

	Eid	EName	EHireDate	ESalary	ECommission
1	1	Sakhawat	2016	10000	500
2	11	Hamza	2021	10000	500
3	21	Osama	2018	15000	600
4	31	Raza	2019	20000	700
5	41	Haris	2020	10000	800

SQLQuery1.sql - D:\U8TAPMM\Del1 (62)* - X

```

create table Employee (Eid int primary key not null identity (1,10), EName varchar(Max), EHireDate varchar(MAX), ESalary int, ECommission int);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Sakhawat', 2016, 50000, 500);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Hamza', 2017, 50000, 500);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Osama', 2018, 40000, 600);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Raza', 2019, 30000, 700);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Haris', 2020, 20000, 800);
select * from Employee;
update Employee set ESalary = 25000;
select * from Employee;

```

100 %

Results Messages

	Eid	EName	EHireDate	ESalary	ECommission
1	1	Sakhawat	2016	25000	500
2	11	Hamza	2021	25000	500
3	21	Osama	2018	25000	600
4	31	Raza	2019	25000	700
5	41	Haris	2020	25000	800

SQLQuery1.sql - g-m-gTAPMM\Del1 (62)* - X

```

create table Employee (Eid int primary key not null identity (1,10), EName varchar(Max), EHireDate varchar(MAX), ESalary int, ECommission int);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Sakhawat', 2016, 50000, 500);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Hamza', 2017, 50000, 500);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Osama', 2018, 40000, 600);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Raza', 2019, 30000, 700);
insert into Employee (EName, EHireDate, ESalary, ECommission) values ('Haris', 2020, 20000, 800);
update Employee set ESalary = 25000;
select * from Employee;

```

100 % • 4

@ Results @ Messages

	Eid	EName	EHireDate	ESalary	ECommission
1	1	Sakhawat	2016	25000	500
2	11	Hamza	2021	25000	500
3	21	Osama	2018	25000	600
4	31	Raza	2019	25000	700
5	41	Haris	2020	25000	800

```
SQLQuery1.sql - D:\U8TAPMM\De11 (62)) * X
create table Employee (Eid int primary key not null identity (1,10), EName varchar(Max), EHireDate varchar(MAX), ESalary int, ECommission int);
insert into Employee (Ename, EHireDate, ESalary, ECommission) values ('Sakhawat', 2016, 50000, 500);
insert into Employee (Ename, EHireDate, ESalary, ECommission) values ('Hamza', 2017, 50000, 500);
insert into Employee (Ename, EHireDate, ESalary, ECommission) values ('Osama', 2018, 40000, 600);
insert into Employee (Ename, EHireDate, ESalary, ECommission) values ('Raza', 2019, 30000, 700);
insert into Employee (Ename, EHireDate, ESalary, ECommission) values ('Haris', 2020, 20000, 800);
update Employee set EHireDate = 2015 where Ename = 'Raza';
select * from Employee;
update Employee set ESalary = 16000 where Ename = 'haris';
update Employee set ESalary = 36000 where Ename = 'Sakhawat';
select * from Employee;
update Employee set ECommission = 1500;
select * from Employee;
delete from Employee where ESalary = 16000;
```

100 %

Results Messages

	Eid	EName	EHireDate	ESalary	ECommission
1	1	Sakhawat	2016	36000	1500
2	11	Hamza	2021	25000	1500
3	21	Osama	2018	25000	1500
4	31	Raza	2015	25000	1500

LAB # 05

TASK 1:

Consider the following relation EMPLOYEE (EMP_ID, EMP_NAME, EMP_ADDRESS, SKILL, PROJ-ID).
EQUIPMENT (EQP-ID, EMP_ID, EQP-TYPE, PROJECT).

- Find the join of relations EMPLOYEE and EQUIPMENT.
 - Get all employees for projects using EQP-TYPE as a “Welding machine”.
 - Get all machines being used at the Mumbai Project?
 - Find all employees of the project using equipment number 110.
-

```
-Create Table Emp ('Empid int primary key identity::1,1', Empname varchar::100, Empaddress varchar::100, Skill varchar::100, Projectid int.);
```

```
insert into Emp:Empname, Empaddress, Skill, Projectid' . values: "Asghar", 'Jauhar', 'C++', 204',
insert into Emp:Empname, Empaddress, Skill, Projectid' . values: "Shafiq", 'Manipur', 'Assembly Language', 205',
insert into Emp:Empname, Empaddress, Skill, Projectid' . values: "Rehman", 'DHA', 'HTML', 206::;
insert into Emp:Empname, Empaddress, Skill, Projectid' . values: "Kamran", 'Bahadurabad', 'CSS', 207::;
insert into Emp:Empname, Empaddress, Skill, Projectid' . values: "Adil", 'Gizni', 'Android Developer', 208 ;
```

```
select ' from Emp:
```

Empid	Empname	Empaddress	Skill	Projectid
1	Asghar	Jauhar	C++	204
2	Shafiq	Manipur	Assembly Language	205
3	Rehman	DHA	HTML	206
4	Kamran	Bahadurabad	CSS	207
5	Adil	Gizni	Android Developer	208

```
-Create Table Equipment:Eqpid int primary key identity::1,1', Empid int, Eqtype varchar::50, Project varchar::50',
FOREIGN KEY :: Empid :: REFERENCES Emp:: Empid :: ;
```

```
insert into Equipment values: '2, 'Welding machine', 'Mumbai' ',
insert into Equipment values: '4, 'Dumper', 'Waapur' ',
insert into Equipment values: '1, 'Bulldozer', 'Dehli' ',
insert into Equipment values: '5, 'Crane', 'Dehli' ,
```

```
select ' from Equipment:
```

Eqpid	Empid	Eqtype	Project
1	2	Welding machine	Mumbai
2	4	Dumper	Kanpur
3	1	Bulldozer	Dehli
4	5	Crane	Dehli

```
Select Emp.Projectid From Emp Inner Join Equipment On Emp.Empid=Equipment.Eqpid
Select Emp.Projectid From Emp Left Join Equipment On Emp.Empid=Equipment.Eqpid
Select Emp.Projectid From Emp Right Join Equipment On Emp.Empid=Equipment.Eqpid
Select Emp.Projectid, Emp.Empname From Emp Full Outer Join Equipment On Equipment.Eqpid=Equipment.Eqpid
```

Projectid	Empname
1	204
2	205
3	206
4	207
13	206
14	207
15	208
16	204
17	205
18	206
19	207
20	208

```
Select * From Equipment Where Eqptype = 'Welding Machine'
Select * From Equipment Where Project = 'Mumbai'
Select * From Equipment Where Eqpid = 2
```

110 %

Results Messages

	Eqpid	Empid	Eqptype	Project
1	1	2	Welding machine	Mumbai

	Eqpid	Empid	Eqptype	Project
1	1	2	Welding machine	Mumbai

	Eqpid	Empid	Eqptype	Project
1	2	4	Dumper	Kanpur

TASK2:

Consider the following relation SALESMAN (SALESMAN_ID, NAME, CITY, COMMISSION) CUSTOMER (CUSTOMER_ID, CUST_NAME, CITY, GRADE, SALESMAN_ID) ORDERS (ORD_NO, PURCH_AMT, ORD_DATE, CUSTOMER_ID, SALESMAN_ID).

- Write a SQL statement to know which salesman are working for which customer.
- Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.
- Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customers or not yet joined under any of the customers.
- Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for those customers from the existing list who placed one or more orders or which order(s) have been placed by the customer who are not in the list.

```
Create Table Salesman (Sid Int Primary Key Not Null Identity (1,1),Sname Nvarchar (Max) ,Scity Nvarchar (Max) ,
Scommision Nvarchar (Max));
```

```
Insert Into Salesman Values ('Hunaid' , 'Hyd' , 55);
Insert Into Salesman Values ('Shakir' , 'Khi' , 65);
Insert Into Salesman Values ('Raza' , 'Lhr' , 78);
```

```
Select * From Salesman;
```

110 %

Results Messages

	Sid	Sname	Scity	Scommision
1	1	Hunaid	Hyd	55
2	2	Shakir	Khi	65
3	3	Raza	Lhr	78

```

Create Table Customer (Cid Int Primary Key Not Null Identity (1,1),Custname Nvarchar (Max) ,Custcity Nvarchar(Max) ,
Custgrade Nvarchar (Max) ,
Sid Int Foreign Key References Salesman (Sid));

Insert Into Customer Values ('Ahad' , 'Fsd','C' , '4');
Insert Into Customer Values ('Rizwan' , 'Khi','F' , '1');
Insert Into Customer Values ('Bilal' , 'Fsd','B' , '5');
Insert Into Customer Values ('Kashif' , 'Khi','A' , '5');
Select * From Customer;

```

110 %

Results Messages

	Cid	Custname	Custcity	Custgrade	Sid
1	2	Rizwan	Khi	F	1

```

Create Table Order_ (Oid Int Primary Key Not Null Identity (1,1), Ordno Int , Ord purch_Amt Int,Orddate Date,
Sid Int Foreign Key References Salesman (Sid),Cid Int Foreign Key References Customer (Cid));

Insert Into Order_ Values (22,2200,3,1, '2020-10-15');
Insert Into Order_ Values (24,2500,3,2, '2020-04-22');
Insert Into Order_ Values (26,300,15,3, '2019-11-02')

Select * From Order_;

```

110 %

Results Messages

	Oid	Ordno	Ord purch_Amt	Orddate	Sid	Cid
--	-----	-------	---------------	---------	-----	-----

LAB # 06

TASK 1:

1. Group the employees by their salaries.
2. List hiredates in descending order.
3. List hiredates in ascending order
4. List all employees and add 20 rupees in each salary.
5. List all 'manager' and add 100 rupees in each salary.
6. List all 'salesman and add 500 rupees in each salary.
7. List all 'clerk' and add 50 rupees in each salary.
8. Find eight percent of salesmen salary.
9. Find the annual salary of each employee.
10. Find the six-month salary of each employee.
11. Find the two-month salary of each manager.
12. Make a query in which all the arithmetic expressions will include all the result will be remaining same.
13. Find the daily wages of each employee.
14. Find the daily wages of each 'manager'.
15. Display your name in lowercase.
16. Find the first and second characters of enames.

17. Find the minimum salary from the table emp.
18. Find the maximum salary from the table emp.
19. Find the length of all ename.
20. Find the length of job.
21. Find the sum of all salaries.
22. Find those employees whose department location is Newark.
23. Find those employees who are working in accounting department.
24. Count all employees.
25. Display the sum of all employees' salaries.
26. How many managers do we have?
27. How many departments do we have?
28. List average salary of each job.
29. Find the Maximum and Minimum salary of all employees.
30. Find the average salaries of those employees who work in dept 10.
31. Find average and sum of all the salaries of each job excluding clerks.
32. Find the minimum and average salary of each department excluding deptno 10.

```

SQLQuery1.sql - D:\U8TAPMM\Del (52)*
CREATE TABLE EMPLOYEEESS(EMP_ID INT, FIRST_NAME VARCHAR(100), LAST_NAME VARCHAR(100), EMAIL VARCHAR(100), PHONE_NUMBER VARCHAR(100),
HIRE_DATE VARCHAR(100), JOB_ID VARCHAR(100), SALARY INT, COMMISSION INT, MANAGER_ID INT, DEPT_ID INT ,DEPTNAME VARCHAR(100),
DESIGNATION VARCHAR(100), LOCATION VARCHAR(100));

INSERT INTO EMPLOYEEESS(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER , HIRE_DATE, JOB_ID, SALARY, COMMISSION, MANAGER_ID,
DEPT_ID,DEPTNAME, DESIGNATION, LOCATION)
VALUES(100, 'LEX', 'KING', 'L KING', 515123, '17-06-2000', 'AD_PRES', 24000, 0.00, 0, NULL,'ACCOUNTS','MANAGER','NEWARK');

INSERT INTO EMPLOYEEESS(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER , HIRE_DATE, JOB_ID, SALARY, COMMISSION,
MANAGER_ID, DEPT_ID,DEPTNAME, DESIGNATION, LOCATION)
VALUES(101, 'STEVEN', 'KOCHHAR', 'STKOCH', 5151234568, '18-06-2000', 'AD_VP', 17000, 0.00, 100, 90,'FINANCE','SALESMAN','GPS');

INSERT INTO EMPLOYEEESS(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER , HIRE_DATE, JOB_ID, SALARY, COMMISSION,
MANAGER_ID, DEPT_ID,DEPTNAME, DESIGNATION, LOCATION)
VALUES(102, 'NEENA', 'DEHAAN', 'NEEDA', 5151234569, '20-06-2000', 'AD_VP', 9000, 0.00, 102, 60,'ACCOUNTS','CLERK','NEWARK');

INSERT INTO EMPLOYEEESS(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER , HIRE_DATE, JOB_ID, SALARY, COMMISSION,
MANAGER_ID, DEPT_ID,DEPTNAME, DESIGNATION, LOCATION)
VALUES(103, 'ALEXANDER', 'HUNOLD', 'ALHUN', 5151234570, '25-06-2000', 'IT_PROG', 6000, 0.00, 101, 60,'FINANCE','MANAGER','GPS');

INSERT INTO EMPLOYEEESS(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER , HIRE_DATE, JOB_ID, SALARY, COMMISSION,
MANAGER_ID, DEPT_ID,DEPTNAME, DESIGNATION, LOCATION)
VALUES(104, 'DAVID', 'AUSTIN', 'DAUSTIN', 5151234572, '19-06-2000', 'IT_PROG', 4800, 0.00,103, 60,'FINANCE','CLERK','GPS');

SELECT * FROM EMPLOYEEESS;

```

	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION	MANAGER_ID	DEPT_ID	DEPTNAME	DESIGNATION	LOCATION
1	100	LEX	KING	L KING	515123	-1989	AD_PRES	24000	0	0	NULL	ACCOUNTS	MANAGER	NEWARK
2	101	STEVEN	KOCHHAR	STKOCH	5151234568	18-06-2000	AD_VP	17000	0	100	90	FINANCE	SALESMAN	GPS
3	102	NEENA	DEHAAN	NEEDA	5151234569	20-06-2000	AD_VP	9000	0	102	60	ACCOUNTS	CLERK	NEWARK
4	103	ALEXANDER	HUNOLD	ALHUN	5151234570	25-06-2000	IT_PROG	6000	0	101	60	FINANCE	MANAGER	GPS
5	104	DAVID	AUSTIN	DAUSTIN	5151234572	19-06-2000	IT_PROG	4800	0	103	60	FINANCE	CLERK	GPS

1) `select SALARY, count(SALARY) from EMPLOYEESS group by SALARY;`

	SALARY	(No column name)
1	4800	1
2	6000	1
3	9000	1
4	17000	1
5	24000	1

2) `SELECT * FROM EMPLOYEESS ORDER BY HIRE_DATE DESC;`

	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION	MANAGER_ID	DEPT_ID
1	103	ALEXANDER	HUNOLD	ALHUN	5151234570	25-06-2000	IT_PROG	6000	0	101	60
2	102	NEENA	DEHAAN	NEEDA	5151234569	20-06-2000	AD_VP	9000	0	102	60
3	101	STEVEN	KOCHHAR	STKOCH	5151234568	18-06-2000	AD_VP	17000	0	100	90
4	100	LEX	KING	LKING	515123	-1989	AD_PRES	24000	0	0	NULL
5	104	DAVID	AUSTIN	DAUSTIN	5151234572	19-06-2000	IT_PROG	4800	0	103	60

3) `SELECT * FROM EMPLOYEESS ORDER BY HIRE_DATE ASC;`

	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION	MANAGER_ID	DEPT_ID
1	104	DAVID	AUSTIN	DAUSTIN	5151234572	19-06-2000	IT_PROG	4800	0	103	60
2	100	LEX	KING	LKING	515123	-1989	AD_PRES	24000	0	0	NULL
3	101	STEVEN	KOCHHAR	STKOCH	5151234568	18-06-2000	AD_VP	17000	0	100	90
4	102	NEENA	DEHAAN	NEEDA	5151234569	20-06-2000	AD_VP	9000	0	102	60
5	103	ALEXANDER	HUNOLD	ALHUN	5151234570	25-06-2000	IT_PROG	6000	0	101	60

4) `SELECT SALARY+20 FROM EMPLOYEESS;`

	(No column name)
1	24020
2	17020
3	9020
4	6020
5	4820

5) `SELECT SALARY+100 FROM EMPLOYEESS WHERE DESIGNATION = 'Manager';`

	(No column name)
1	24100
2	6100

6) `SELECT SALARY+500 FROM EMPLOYEESS WHERE DESIGNATION = 'Salesman';`

(No column name)	
1	17500

7) `SELECT SALARY+50 FROM EMPLOYEESS WHERE DESIGNATION = 'Clerk';`

(No column name)	
1	9050
2	4850

8) `SELECT SALARY/8 as '8th Percent' FROM EMPLOYEESS WHERE DESIGNATION = 'Salesman';`

8th Percent	
1	2125

9) `SELECT SALARY*12 as 'Annual_Salary' FROM EMPLOYEESS;`

Annual_Salary	
1	288000
2	204000
3	108000
4	72000
5	57600

10) `SELECT SALARY*6 as '6_Month_Salary' FROM EMPLOYEESS;`

6_Month_Salary	
1	144000
2	102000
3	54000
4	36000
5	28800

11) `SELECT SALARY*2 as '2_Month_Salary' FROM EMPLOYEESS;`

Results		Messages	
2_Month_Salary			
1	48000		
2	34000		
3	18000		
4	12000		
5	9600		

12) **SELECT** SALARY*9, SALARY-2,SALARY/3,SALARY+450 **FROM** EMPLOYEESS;

	(No column name)	(No column name)	(No column name)	(No column name)
1	216000	23998	8000	24450
2	153000	16998	5666	17450
3	81000	8998	3000	9450
4	54000	5998	2000	6450
5	43200	4798	1600	5250

13) **SELECT** SALARY/30 as 'Daily Wages' **FROM** EMPLOYEESS;

	Daily Wages
1	800
2	566
3	300
4	200
5	160

14) **SELECT** SALARY/30 as 'Daily Wages' **FROM** EMPLOYEESS **WHERE** DESIGNATION = 'Manager';

	Daily Wages
1	800
2	200

15) **SELECT** LOWER('UZAIR');

	(No column name)
1	uzair

16) **select** substring(FIRST_NAME,1,2)**from** EMPLOYEESS;

	(No column name)
1	LE
2	ST
3	NE
4	AL
5	DA

17) **SELECT** MIN(SALARY) **FROM** EMPLOYEES;

	(No column name)
1	200

18) `SELECT MAX(SALARY) AS 'MAX SALARY' FROM EMPLOYEEES;`

	MAX SALARY
1	24000

19) `SELECT LEN(FIRST_NAME) AS 'LENGTH' FROM EMPLOYEEES;`

	LENGTH
1	3
2	6
3	5
4	9
5	5

20) `SELECT LEN(DSIGNATION) FROM EMPLOYEEES;`

	(No column name)
1	7
2	8
3	5
4	7
5	5

21) `SELECT SUM(SALARY) AS 'SUM' FROM EMPLOYEEES;`

	SUM
1	60800

22) `SELECT * FROM EMPLOYEEES WHERE LOC = 'Newark';`

	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION	MANAGER_ID	DEPT_ID	DEPTNAME	DESIGNATION	LOCATION
1	100	LEX	KING	LKING	515123	-1989	AD_PRES	24000	0	0	NULL	ACCOUNTS	MANAGER	NEWARK
2	102	NEENA	DEHAAN	NEEDA	5151234569	20-06-2000	AD_VP	9000	0	102	60	ACCOUNTS	CLERK	NEWARK

23) `SELECT * FROM EMPLOYEEES WHERE DEPT_NAME = 'Accounts';`

	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION	MANAGER_ID	DEPT_ID	DEPTNAME	DESIGNATION	LOCATION
1	100	LEX	KING	LKING	515123	-1989	AD_PRES	24000	0	0	NULL	ACCOUNTS	MANAGER	NEWARK
2	102	NEENA	DEHAAN	NEEDA	5151234569	20-06-2000	AD_VP	9000	0	102	60	ACCOUNTS	CLERK	NEWARK

24) `SELECT COUNT(EMP_ID) FROM EMPLOYEEES;`

	(No column name)
1	5

25) `SELECT COUNT(DESIGNATION) FROM EMPLOYEEES WHERE DESIGNATION = 'Manager';`

	(No column name)
1	2

26) `SELECT COUNT(DEPT_NAME) FROM EMPLOYEEES;`

	(No column name)
1	5

27) `SELECT AVG(SALARY) as 'Managers Average Salary' FROM EMPLOYEEES WHERE DESIGNATION = 'Manager';`

	Managers Average Salary
1	15000

28) `SELECT AVG(SALARY) as 'Salemsans Average Salary' FROM EMPLOYEEES WHERE DESIGNATION = 'Salesman';`

	Salemsans Average Salary
1	17000

29) `SELECT AVG(SALARY) as 'Clerks Average Salary' FROM EMPLOYEEES WHERE DESIGNATION = 'Clerk';`

	Clerks Average Salary
1	6900

30) `SELECT AVG(SALARY) as 'Analysts Average Salary' FROM EMPLOYEEES WHERE DESIGNATION = 'Analyst';`

	Analysts Average Salary
1	NULL

31) `SELECT AVG(SALARY) FROM EMPLOYEEES WHERE DEPT_ID = 10;`

	(No column name)
1	NULL

32) `SELECT AVG(SALARY), SUM(SALARY) FROM EMPLOYEEES WHERE DESIGNATION != 'Clerk';`

	AVG SALARY	SUM SALARY
1	15666	47000

33) `SELECT MIN(SALARY) AS 'MIN SALARY', AVG(SALARY) AS 'AVG SALARY' FROM EMPLOYEEES WHERE`

DEPT_ID != 10;

	MIN SALARY	AVG SALARY
1	4800	9200

LAB # 07

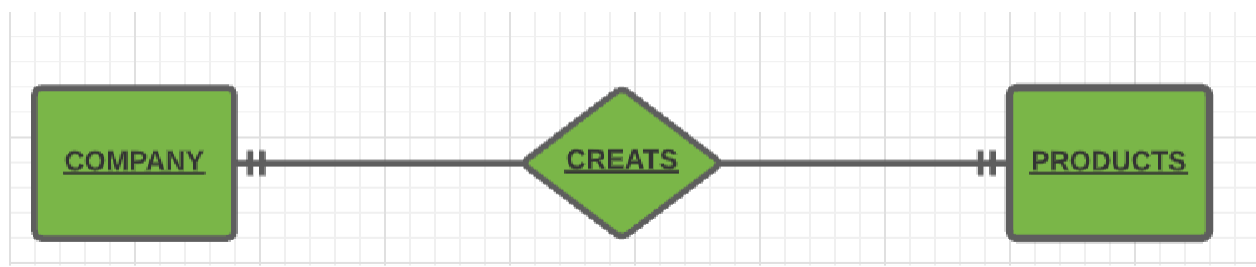
TASK 1:

For a binary relationship set the mapping cardinality must be one of the following types:

1. One to one
2. One to many
3. Many to one
4. Many to many

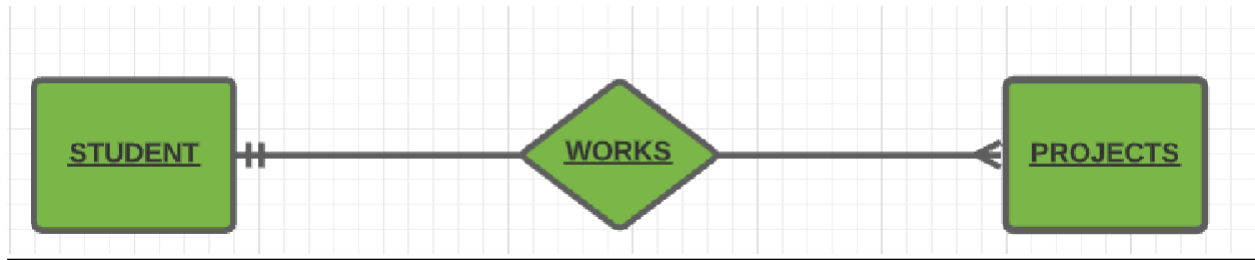
TASK: Implement mapping of tables from theory classes.

1) One-to-One Relationship:



- A Company creates only one Product, and that Product is only made by that Company.

2) One-to-Many Relationship:



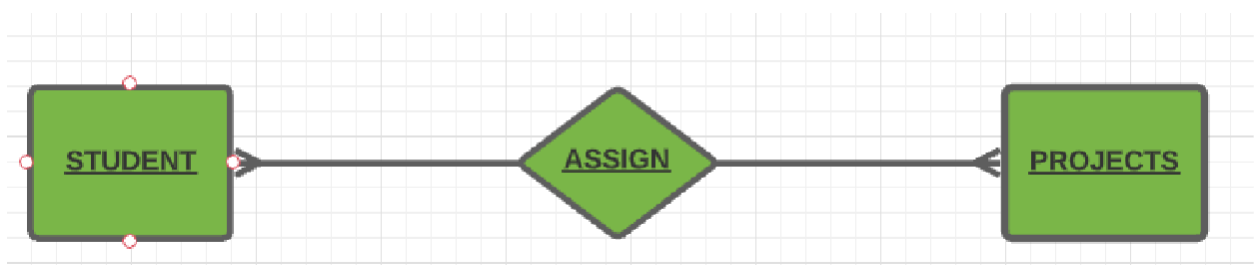
- One Student works on many Projects.

3) Many-to-One Relationship:



- Many Students Can Study in a single College but a Student Cannot Study in many Colleges at the same time.

4) Many-to-Many Relationship:



- A Student can be assigned to many Projects and a Project can be assigned to many Students.

LAB # 08

TASKS:

- 1) Write a query to display the employee name and hiredate for all employees in the same department as Blake. Exclude Blake.
- 2) Create a query to display the employee number and name for all employees who earn more than the average salary.
- 3) Write a query to display the employee number and name for all employees who work in a department with any employee whose name contains a T.
- 4) Display the employee name, department number, and job title for all employees whose department location is Dallas.
- 5) Display the employee name and salary of all employees who report to King.
- 6) Write a query to display the employee name, salary, deptno and job for all employees in the same job as empno 7369.
- 7) Display the employee number, name and salary for all employees who earn more than the average salary and who work in department with any employee with a T in their name.

```
SQLQuery1.sql - D:\U8TAPMM\Del1 (52)* X
create table employeeY (emp_name varchar(20),hiredate varchar(20),emp_number int,salary int,department varchar(20),
depart_number int,job_title varchar(20),depart_loc varchar(20),report varchar(20));
select * from employeeY;
insert into employeeY values('HIRA','01/01/2000',4532,10000,'Blake',213,'graphic designer','newtown','Prince');
insert into employeeY values('MAHA','02/02/2001',1872,20000,'Flake',566,'web designer','dallas','Queen');
insert into employeeY values('ZOOFI','03/03/2002',9413,30000,'Drake',238,'IT manager','newyork','King');
insert into employeeY values('AREEBA','04/04/2003',7369,40000,'Srake',763,'sql development','chicago','King');
insert into employeeY values('SAKAWAT','05/05/2004',1555,50000,'Trake',983,'python dev','dallas','Slave');

select emp_name,hiredate from employeeY where department='Blake';
select emp_number,emp_name from employeeY where salary > (select AVG(salary) from employeeY) order by salary;
select emp_number,emp_name from employeeY where department in (select department from employeeY where emp_name like '%T%');
select emp_name,depart_number,job_title from employeeY where depart_loc = 'Dallas';
select emp_name,salary from employeeY where report = 'King';
select emp_name,salary,depart_number,job_title from employeeY where emp_number = 7369;
select emp_number,emp_name,salary from employeeY where salary > (select AVG(salary) from employeeY) and
department in ( select department from employeeY where emp_name like '%T%');
```

100 %									
Results Messages									
emp_name hiredate emp_number salary department depart_number job_title depart_loc report									
emp_name hiredate									
1	HIRA	01/01/2000							
emp_number emp_name									
1	7369	AREEBA							
2	1555	SAKAWAT							
emp_number emp_name									
1	1555	SAKAWAT							
emp_name depart_number job_title									
1	MAHA	566	web designer						
2	SAKAWAT	983	python dev						
emp_name salary									
1	ZOOFI	30000							
2	AREEBA	40000							
emp_name salary depart_number job_title									
1	AREEBA	40000	763	sql development					
emp_number emp_name salary									
1	1555	SAKAWAT	50000						
Query executed successfully.									

LAB # 09

TASK:

- 1) To display the employee name, department name, and location of all employees who earn a commission.
- 2) To display all the employee's name (including KING who has no manager) and their manager name
- 3) To display the name of all employees whose manager is KING.
- 4) Create a unique listing of all jobs that in department 30. Include the location of department 30 in the Output.
- 5) Write a query to display the name, job, department number and department name for all employees who work in New York
- 6) Display the employee name and employee number along with their manager's name Manager Number. Label the columns Employee, Emp#, Manager, and Manager#, respectively

SQLQuery1.sql - D:\U8TAPMM\ Dell (52)* - X

```
CREATE TABLE EMPLOYEEES(EMP_ID INT, FIRST_NAME VARCHAR(100), LAST_NAME VARCHAR(100), EMAIL VARCHAR(100),  
PHONE_NUMBER VARCHAR(100), HIRE_DATE VARCHAR(100), JOB_ID VARCHAR(100), SALARY INT, COMMISSION INT, MANAGER_ID INT, DEPT_ID INT,  
DEPTNAME VARCHAR(100), DESIGNATION VARCHAR(100), LOCATION VARCHAR(100));  
  
INSERT INTO EMPLOYEEES(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION, MANAGER_ID,  
DEPT_ID, DEPTNAME, DESIGNATION, LOCATION)  
VALUES(100, 'LEX', 'KING', 'L KING', 515123, 17-06-2000, 'AD_PRES', 24000, 0.00, 0, NULL, 'ACCOUNTS', 'MANAGER', 'NEWARK');  
  
INSERT INTO EMPLOYEEES(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION, MANAGER_ID,  
DEPT_ID, DEPTNAME, DESIGNATION, LOCATION)  
VALUES(101, 'STEVEN', 'KOCHHAR', 'ST KOCH', 5151234568, '18-06-2000', 'AD_VP', 17000, 0.00, 100, 90, 'FINANCE', 'SALESMAN', 'GPS');  
  
INSERT INTO EMPLOYEEES(EMP_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION, MANAGER_ID,  
DEPT_ID, DEPTNAME, DESIGNATION, LOCATION)  
VALUES(102, 'NEENA', 'DEHAAN', 'NEEDA', 5151234569, '20-06-2000', 'AD_VP', 9000, 0.00, 102, 60, 'ACCOUNTS', 'CLERK', 'NEWARK');  
  
SELECT * FROM EMPLOYEEES;  
  
CREATE TABLE DEPARTMENT(DEPT_ID INT, DEPTNAME VARCHAR(100), MANAGER_ID INT, LOCATION_ID INT);  
  
INSERT INTO DEPARTMENT(DEPT_ID, DEPTNAME, MANAGER_ID, LOCATION_ID )  
VALUES('10', 'ADMINISTRATION', '200', '1700');  
INSERT INTO DEPARTMENT(DEPT_ID, DEPTNAME, MANAGER_ID, LOCATION_ID )  
VALUES('20', 'MARKETING', '200', '1700');  
INSERT INTO DEPARTMENT(DEPT_ID, DEPTNAME, MANAGER_ID, LOCATION_ID )  
VALUES('30', 'PURCHASING', '200', '1700');  
SELECT * FROM DEPARTMENT;
```

```
CREATE TABLE LOCATIONS(LOCATION_ID INT, STREET_ADDRESS VARCHAR(MAX), POSTAL_CODE INT, CITY VARCHAR(MAX),  
STATE_PROVINCE VARCHAR(MAX), COUNTRY_ID VARCHAR(100));  
  
INSERT INTO LOCATIONS(LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID )  
VALUES('1000', '1297 Via Cola di Rie', '989', 'ROMA', '', 'IT');  
  
INSERT INTO LOCATIONS(LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID )  
VALUES('1100', '93091 Calle della Testa', '10934', 'VENICE', '', 'IT');  
  
INSERT INTO LOCATIONS(LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID )  
VALUES('1200', '2017 Shinjuku-ku', '1689', 'TOKYO', 'TOKYO PREFECTURE', 'JP');  
  
SELECT * FROM LOCATIONS;
```

- 1) **SELECT** e.last_name, d.department_name, d.location_id, l.city
FROM employees e, departments d, locations l
WHERE e.department_id = d.department_id
AND
d.location_id = l.location_id
AND e.commission IS NOT NULL;

- 2) `SELECT w.last_name "Employee", w.employee_id "EMP#",
m.last_name "Manager", m.employee_id "Mgr#"

FROM employees w

LEFT OUTER JOIN employees m

ON (w.manager_id = m.employee_id);`
- 3) `SELECT e.ENAME FROM EMPLOYEES e WHERE EMPLOYEES.DESIGNATION='Manager'
And e.ENAME='King'`
- 4) `SELECT DISTINCT job_id, location_id
FROM employees, departments

WHERE employees.department_id = departments.department_id

AND employees.department_id = 80;`
- 5) `SELECT e.last_name, e.job_id, e.department_id, d.department_name
FROM employees e JOIN departments d

ON (e.department_id = d.department_id)

JOIN locations l

ON (d.location_id = l.location_id)

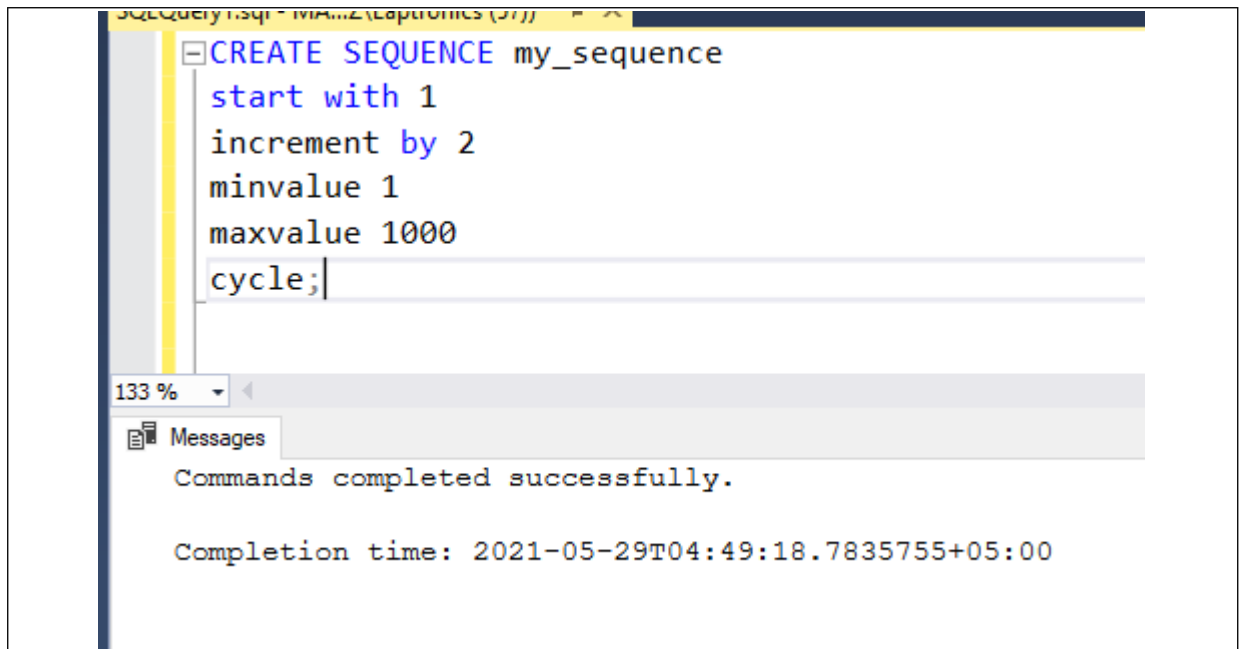
WHERE l.city = 'newyork';`
- 6) `SELECT w.last_name "Employee", w.employee_id "EMP#",
m.last_name "Manager", m.employee_id "Mgr#"
FROM employees w join employees m

ON (w.manager_id = m.employee_id);`

LAB # 10

TASKS:

- 1) CREATE SEQUENCE my_sequence MINVALUE 1 MAXVALUE 1000 START WITH 1 INCREMENT BY 2;
- 2) Create a new table Person with my_sequence and insert at least 5 records.
- 3) Create a new table Customer with my_sequence and insert at least 5 records
- 4) Create B-Tree indexes on
 - Name column of EMP table
 - Designation column of EMP table
 - First 10 characters of Title in TRAININGtable
- 5) Create bitmapped indexes on
 - Gender column of EMP table
 - Performance column of EMP_PROJECT table



The screenshot displays the SQL Developer interface. The top pane shows the SQL command: `CREATE SEQUENCE my_sequence start with 1 increment by 2 minvalue 1 maxvalue 1000 cycle;`. The bottom pane, titled 'Messages', shows the output: 'Commands completed successfully.' and 'Completion time: 2021-05-29T04:49:18.7835755+05:00'.

```
SQL> CREATE SEQUENCE my_sequence
      start with 1
      increment by 2
      minvalue 1
      maxvalue 1000
      cycle;
```

133 %

Messages

Commands completed successfully.

Completion time: 2021-05-29T04:49:18.7835755+05:00

```
CREATE TABLE person(  
ID int, NAME char(20)
```

```
INSERT into person VALUES (next value for my_sequence, 'LEE');  
INSERT into person VALUES (next value for my_sequence, 'LEX');  
INSERT into person VALUES (next value for my_sequence, 'STEVEN');  
INSERT into person VALUES (next value for my_sequence, 'REENA');  
INSERT into person VALUES (next value for my_sequence, 'KING');
```

```
(1 row affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

Completion time: 2021-05-29T04:50:26.4103289+05:00

```
CREATE TABLE Customer(  
ID int,  
NAME char(20)  
);  
  
INSERT into Customer VALUES (next value for my_sequence, 'STAN');  
INSERT into Customer VALUES (next value for my_sequence, 'RUTHER');  
INSERT into Customer VALUES (next value for my_sequence, 'ANNA');  
INSERT into Customer VALUES (next value for my_sequence, 'WANNA');  
INSERT into Customer VALUES (next value for my_sequence, 'JOHN');
```

1 %

Messages

```
(1 row affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

Completion time: 2021-05-29T04:52:52.1032434+05:00

LAB # 11

TASKS:

- Consider the following schema, in the form of normalized relations, to represent information about employees, grades, training and projects in an organization.

EMPLOYEE Empno (eg 6712) Name

Designation (e.g. Database Developer) Qualification

Joindate PROJECT PID (eg P812)

Title Client Duration (in weeks)

Status (New, In Progress, Complete) EMP_PROJECT

Empno PID

Performance (Excellent, Good, Fair, Bad, Poor)

GRADE

Designation Grade (1-20) Total

Posts PostsAvailable (<= TotalPosts)

TRAINING

Tcode (eg T902) Title

StartDate

EndDate

EMP_TRAINING

Empno Tcode

Attendance (%)

- 1) Develop a script file EMPLOYEE.SQL to create tables for the above schema. Implement all necessary integrity constraints including primary and foreign keys. (NOTE: All check constraints should be at table level)
- 2) Write SQL statements to add
 - Gender column to EMP table. The only possible values are Male and Female.
 - Instructor_Name column to TRAINING table.
 - Salary column to GRADE table.

- 3) Write down a transaction to insert data in EMP_TRAINING table. The data should be finally saved in the database.
- Employee 3400 gets Developer 6i training and his attendance is 87%
 - Employee 3300 gets Typing/shorthand training and her attendance

```
CREATE TABLE EMPLOYEESS (EMPNO INT PRIMARY KEY IDENTITY(1,1), NAME VARCHAR(30), DESIGNATION VARCHAR(30), QUALIFICATION VARCHAR(50), JOINDATE DATE);
CREATE TABLE GRADE (DESIGNATION VARCHAR(10), GRADE INT CHECK (GRADE >= 1 AND GRADE <= 20), TOTALPOSTS INT, POSTAVAILABLE INT);
CREATE TABLE PROJECT (PID VARCHAR(10) PRIMARY KEY, TITLE VARCHAR(20), CLIENT VARCHAR(20), DURATION INT, STATUS VARCHAR(20));
CREATE TABLE TRAINING (TCODE VARCHAR(10) PRIMARY KEY, TITLE VARCHAR(20), STARTDATE DATE, ENDDATE DATE);
CREATE TABLE EMP_PROJECT (EMPNO INT, PID VARCHAR(10), PERFORMANCE VARCHAR(30), FOREIGN KEY (EMPNO) REFERENCES EMPLOYEE (EMPNO), FOREIGN KEY (PID) REFERENCES PROJECT (PID));
CREATE TABLE EMP_TRAINING (EMPNO INT, TCODE VARCHAR(10), ATTENDANCE INT, FOREIGN KEY (EMPNO) REFERENCES EMPLOYEE (EMPNO), FOREIGN KEY (TCODE) REFERENCES TRAINING (TCODE));
```

100 %

Messages

Commands completed successfully.

Completion time: 2021-05-29T04:56:41.4977942+05:00

LAB # 13

Task:

- 1) Creating a server audit with a file target
- 2) Creating a server audit with a Windows Application log target with options
- 3) Creating a server audit containing a WHERE clause

1) Creating a server audit with a file target

CREATESERVERAUDIT HIPAA_Audit

TOFILE(FILEPATH = '\\SQLPROD_1\\Audit\\');

```
121 %
```

Messages

Commands completed successfully.

Completion time: 2021-05-29T05:54:05.4568505+05:00

2) Creating a server audit with a Windows Application log target with options

```
CREATE SERVER AUDIT HIPAA_Audit
TO APPLICATION_LOG
WITH( QUEUE_DELAY = 1000, ON_FAILURE = SHUTDOWN);
```

121 %

Messages

Commands completed successfully.

Completion time: 2021-05-29T05:58:18.8655996+05:00

3) Creating a server audit containing a WHERE clause

```
SQLQuery1.sql - D...U8TAPMM\ Dell (52)) *  X
CREATEDATABASE TestDB;
GO
USE TestDB;
GO
CREATE SCHEMA DataSchema;
GO
CREATE TABLE DataSchema.GeneralData (ID INT PRIMARY KEY, DataField VARCHAR(50)
NOT NULL);
GO
CREATE TABLE DataSchema.SensitiveData (ID INT PRIMARY KEY, DataField VARCHAR(50)
NOT NULL);
GO
-- Create the server audit in the master database
USE master;
GO
CREATE SERVER AUDIT AuditDataAccess
TO FILE( FILEPATH = 'C:\SQLAudit\' )
WHERE object_name = 'SensitiveData' ;
GO
ALTER SERVER AUDIT AuditDataAccess WITH (STATE = ON);
GO
```

```

-- Create the database audit specification in the TestDB database
USE TestDB;

CREATE DATABASE AUDIT SPECIFICATION [FilterForSensitiveData]
FOR SERVER AUDIT [AuditDataAccess]
ADD (SELECT ON S[HEMA::[DataSchema] BY [public])
WITH (STATE = ON);

-- Trigger the audit event by selecting front tables
SELECT ID, DataField FROM DataSchema.GeneralData;
SELECT ID, DataField FROM DataSchema.SensitiveData;

-- - Check the audit for the filtered content
SELECT FROM fnget_audit_file('C:\SQLAudit\AuditDataAccess *.sqlaudit',default,default);

```