

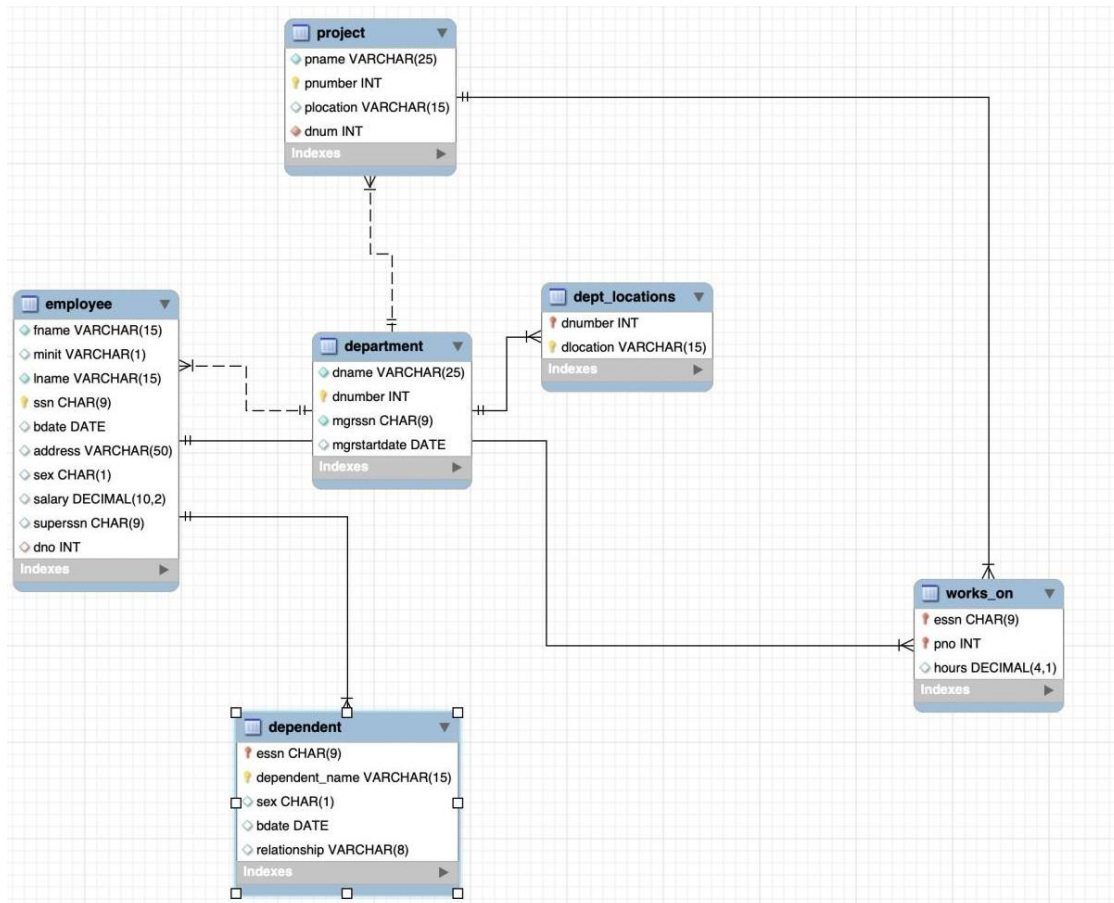
COMP- 8115-M50 Database systems

Quiz – 5

1. [30 pts] Using the MySQL scripts in the companydb.sql file, generate a new company database.

The companydb.sql file can be found on canvas. Put a screenshot of your final database setup.

ER diagram for the final company database sql file.



It has different tables like Project, employees, department, dep_locations, works_on, dependent.

Every tables has their own attributes and they are connected with each other in different ways.

The attributes of tables can be understood from the table schema so let me mention the schema given in the company sql file

```
CREATE TABLE department (  
  dname    varchar(25) not null,  
  dnumber  integer(4),  
  mgrssn   char(9) not null,
```

```
mgrstartdate date,  
primary key (dnumber),  
key (dname)  
);
```

```
CREATE TABLE employee (  
    fname varchar(15) not null,  
    minit varchar(1),  
    lname varchar(15) not null,  
    ssn char(9),  
    bdate date,  
    address varchar(50),  
    sex char,  
    salary decimal(10,2),  
    superssn char(9),  
    dno integer(4),  
    primary key (ssn),  
    foreign key (dno) references department(dnumber)  
);
```

```
CREATE TABLE dept_locations (  
    dnumber integer(4),  
    dlocation varchar(15),  
    primary key (dnumber,dlocation),  
    foreign key (dnumber) references department(dnumber)  
);
```

```
CREATE TABLE project (  
    pname varchar(25) not null,  
    pnumber integer(4),  
    plocation varchar(15),
```

```
dnum    integer(4) not null,  
primary key (pnumber),  
unique (pname),  
foreign key (dnum) references department(dnumber)  
);
```

```
CREATE TABLE works_on (  
    essn char(9),  
    pno integer(4),  
    hours decimal(4,1),  
    primary key (essn,pno),  
    foreign key (essn) references employee(ssn),  
    foreign key (pno) references project(pnumber)  
);
```

```
CREATE TABLE dependent (  
    essn char(9),  
    dependent_name varchar(15),  
    sex char,  
    bdate date,  
    relationship varchar(8),  
    primary key (essn,dependent_name),  
    foreign key (essn) references employee(ssn)  
);
```

The values to these tables are being inserted through insert statement.

Now, let me show the values in different tables

Values for the dependent table

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
3 DROP TABLE IF EXISTS employee;
4 DROP TABLE IF EXISTS dept_locations;
5 DROP TABLE IF EXISTS project;
6 DROP TABLE IF EXISTS works_on;
7 DROP TABLE IF EXISTS dependent;
8 select * from dependent;
9 CREATE TABLE department (
10     dname varchar(25) not null,
11     dnumber integer(4),
12     mgrssn char(9) not null,
13     mgrstartdate date,
14     primary key (dnumber),
15     key (dname)
16 );
```

The result grid shows the data for the 'dependent' table:

essn	dependent_name	sex	bdate	relationship
123456789	Alice	F	1978-12-31	Daughter
123456789	Elizabeth	F	0000-00-00	Spouse
123456789	Michael	M	1978-01-01	Son
333445555	Alice	F	1978-04-05	Daughter
333445555	Joy	F	1948-05-03	Spouse
333445555	Theodore	M	1973-10-25	Son
444444400	Johnny	M	1997-04-04	Son
444444400	Tommy	M	1999-06-07	Son
444444401	Chris	M	1969-04-19	Spouse
444444402	Sam	M	1964-02-14	Spouse
987654321	Abner	M	1932-02-29	Spouse

Values for the Employee table

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
3 DROP TABLE IF EXISTS employee;
4 DROP TABLE IF EXISTS dept_locations;
5 DROP TABLE IF EXISTS project;
6 DROP TABLE IF EXISTS works_on;
7 DROP TABLE IF EXISTS dependent;
8 select * from dependent;
9 select * from employee;
10 CREATE TABLE department (
11     dname varchar(25) not null,
12     dnumber integer(4),
13     mgrssn char(9) not null,
14     mgrstartdate date,
15     primary key (dnumber),
16     key (dname)
17 );
```

The result grid shows the data for the 'employee' table:

fname	minit	lname	ssn	bdate	address	sex	salary	superssn	dno
Jared	D	James	111111100	1966-10-10	123 Peachtree, Atlanta, GA	M	85000.00	null	6
Jon	C	Jones	111111101	1967-11-14	111 Allgood, Atlanta, GA	M	49000.00	111111100	6
Justin	n	Mark	111111102	1966-01-12	2542 May, Atlanta, GA	M	40000.00	111111100	6
Brad	C	Knight	111111103	1968-02-13	126 Main St., Atlanta, GA	M	44000.00	111111100	6
John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5
Evan	E	Wallis	222222200	1958-01-16	134 Pelham, Milwaukee, WI	M	92000.00	null	7
Josh	U	Zell	222222201	1954-05-22	266 McGrady, Milwaukee, WI	M	56000.00	222222200	7
Andy	C	Vile	222222202	1944-06-21	1967 Jordan, Milwaukee, WI	M	53000.00	222222200	7
Tom	G	Brand	222222203	1966-12-16	112 Third St, Milwaukee, WI	M	62500.00	222222200	7
Jerry	F	Vos	222222204	1967-11-11	263 Mayberry, Milwaukee, WI	F	61000.00	222222201	7
Chris	A	Carter	222222205	1960-03-21	565 Jordan, Milwaukee, WI	F	43000.00	222222201	7
Kim	C	Grace	333333300	1970-10-23	6677 Mills Ave, Sacramento, CA	F	79000.00	null	6

Values from the dept_locations

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL code:

```
3 DROP TABLE IF EXISTS employee;
4 DROP TABLE IF EXISTS dept_locations;
5 DROP TABLE IF EXISTS project;
6 DROP TABLE IF EXISTS works_on;
7 DROP TABLE IF EXISTS dependent;
8 select * from dependent;
9 select * from employee;
10 select * from dept_locations;
11
12 CREATE TABLE department (
13     dname varchar(25) not null,
14     dnumber integer(4),
15     mgrssn char(9) not null,
16     mgrstartdate date,
```

The Results grid displays the following data for the `dept_locations` table:

dnumber	dlocation
1	Houston
4	Stafford
5	Bellare
5	Houston
5	Sugarland
6	Atlanta
6	Sacramento
7	Milwaukee
8	Chicago
8	Dallas
8	Miami
8	Philadelphia

Values from the table project

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL code:

```
3 DROP TABLE IF EXISTS employee;
4 DROP TABLE IF EXISTS dept_locations;
5 DROP TABLE IF EXISTS project;
6 DROP TABLE IF EXISTS works_on;
7 DROP TABLE IF EXISTS dependent;
8 select * from dependent;
9 select * from employee;
10 select * from dept_locations;
11 select * from project;
12
13 CREATE TABLE department (
14     dname varchar(25) not null,
15     dnumber integer(4),
16     mgrssn char(9) not null,
```

The Results grid displays the following data for the `project` table:

pname	pnumber	plocation	drum
ProductX	1	Bellare	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4
OperatingSystems	61	Jacksonville	6
DatabasesSystems	62	Birmingham	6
Middleware	63	Jackson	6
InkjetPrinters	91	Phoenix	7
LaserPrinters	92	LasVegas	7

The screenshot displays the MySQL Workbench environment. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window is divided into several panes:

- Navigator:** Shows the database structure for 'companydatabase', including Tables, Views, Stored Procedures, and Functions. The 'company' table is selected.
- SQL Editor:** Contains a SQL query with the following statements:


```

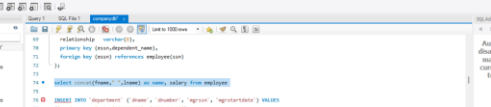
      3 DROP TABLE IF EXISTS employee;
      4 DROP TABLE IF EXISTS dept_locations;
      5 DROP TABLE IF EXISTS project;
      6 DROP TABLE IF EXISTS works_on;
      7 DROP TABLE IF EXISTS dependent;
      8 select * from dependent;
      9 select * from employee;
      10 select * from dept_locations;
      11 select * from project;
      12 select * from works_on;
      13
      14 CREATE TABLE department (
      15     dname varchar(25) not null,
      16     dnumber integer(4),
      
```
- Result Grid:** Displays the results of the query. The first table shows employee data with columns 'esn', 'pno', and 'hours'. The second table shows department data with columns 'dname' and 'dnumber'.

esn	pno	hours
1111111100	61	40.0
1111111101	61	40.0
1111111102	61	40.0
1111111103	61	40.0
123456789	1	32.5
123456789	2	7.5
222222200	62	40.0
222222201	62	40.0
222222202	62	40.0
222222203	62	40.0
222222204	62	40.0
222222205	62	40.0
- Schema:** Shows the selected schema 'companydatabase'.
- Output:** Displays the results of the query execution.

The bottom status bar shows the system clock as 22:33 on 17-06-2022.

Query:

```
select concat(fname," ",lname) as name, salary from employee
```



The screenshot shows a Jupyter Notebook interface. The top bar indicates the file path is `Local untitled.ipynb`. The notebook contains a single code cell with a SQL query. The query is as follows:

```
SELECT
  employee_id,
  last_name,
  salary,
  department_name
FROM
  employees
WHERE
  employee_id IN (
    SELECT
      employee_id
    FROM
      employees
    WHERE
      salary > 12000
  )
```

The query is executed, and the results are displayed in a table. The table has 10 rows and 4 columns: `employee_id`, `last_name`, `salary`, and `department_name`. The results are as follows:

employee_id	last_name	salary	department_name
101	DEHAENE	12000	Finance
102	BAERENTZEN	12000	Finance
103	DEHAENE	12000	Finance
104	DEHAENE	12000	Finance
105	DEHAENE	12000	Finance
106	DEHAENE	12000	Finance
107	DEHAENE	12000	Finance
108	DEHAENE	12000	Finance
109	DEHAENE	12000	Finance
110	DEHAENE	12000	Finance

The bottom of the screenshot shows the Jupyter Notebook interface with a file explorer on the left and a toolbar at the bottom.

```
select concat(emp.fname," ",emp.lname) as name, emp.address from employee as emp join
Department as dept on emp.dno = dept.dnumber where dept.dname = 'Sales'
```

since it is not necessary I am not displaying the output screens

c. [5 pts] Retrieve the name and address of all employees who work either for the 'Sales' or 'Administration' department.

Query:

```
select concat(emp.fname," ",emp.lname) as name, emp.address from employee as emp join
Department as dept on emp.dno = dept.dnumber where dept.dname = 'Sales' or dept.dname =
'Administration'
```

d. [5 pts] Retrieve the name and address of all employees who work either for the 'Sales' or 'Administration' department, sorted alphabetically by last name, then first name.

Query:

```
select emp.fname,emp.lname, emp.address from employee as emp join Department as dept on
emp.dno = dept.dnumber where dept.dname = 'Sales' or dept.dname = 'Administration' order by
emp.lname, emp.fname
```

e. [5 pts] Retrieve all employees in the research department whose salary is between \$30,000 and \$40,000.

Query:

```
Select emp.* from Employee as emp join Department as dept on emp.dno = dept.dnumber Where
emp.Salary between 30000 and 40000 and dept.dname = 'research'
```

f. [5 pts] List all name of employees, the projects that they have been working on, and the amount of time they spent for those projects.

Query:

```
Select concat(emp.fname," ",emp.lname) as name, emp.address, pr.pname, wo.hours from
Employee as emp join Works_on as wo on emp.ssn = wo.essn join Project as pr on wo.pno =
pr.pnumber
```

g. [5 pts] Retrieve all employees whose address is in Atlanta, GA.

Query:

Select * from Employee where address = 'Atlanta, GA'

h. [5 pts] List all the department names which are located in Houston.

Query:

Select dept.*,dl.dlocation from Department as dept join Dept_locations as dl on dept.dnumber = dl.dnumber where dl.dlocation = 'Houston'

i. [5 pts] Retrieve all distinct salary values of every employee.

Query:

Select distinct(Salary), concat(fname," ",lname) as name, address,ssn, bdate from Employee

j. [5 pts] A new person whose name is Joshua T Miller has been hired. Add this new

employed person to the system along with his following information:

SSN: 456363313

birth date: 1971-10-18

address: 98 Oak Forest, Katy, TX

salary: 37000

superssn: 456363313

dno: 4

Query:

Insert into Employee values('Joshua','T','Miller','456363313','1971-10-18','98 Oak Forest, Katy, TX','M','37000','456363313',4)

k. [5 pts] Joshua's salary has been increased with 10%. Make the required update.

Query:

Update Employee set Salary = Salary + Salary *0.1 where fname = 'Joshua'

l. [5 pts] Joshua left the company after working 3 years. Delete his personal record from the system.

Query:

delete Employee where essn = '456363313'

m. [10 pts] Make a list of all project numbers and project names for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

Query:

Select pr.pnumber,pr.pname,emp.lname,dept.dnumber,dept.dname from EMPLOYEE as emp join department dept join project as pr on dept.dnumber = pr.dnum where lname = 'Smith';