1DV503/1DT903 Database Technology and Modeling

Emil Ulvagården
School of Computer Science, Physics
and Mathematics, Linnaeus University, Sweden
eu222dq@student.lnu.se

Task 1

A) {Manufacturer, Serial_number} → Model, Batch, Capacity, Retailer
B) Model_number → Manufacturer
C) Batch → Model

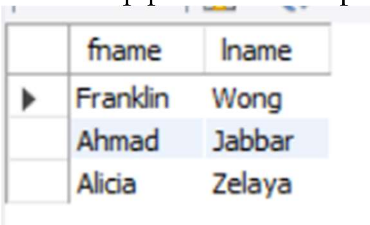
Task 2

The relations is in 1NF due to the fact that it has a primary key that uniquely identifies every record. The Primary key also contains the single valued and non, nested relations. The relation is not in 2NF since one attribute is not dependent on both parts of the primary key. An example of this is commission which is only dependent on the salesperson and not the car. Because the relation is not 2NF it can't be 3NF and the relations can't have a nonkey attribute depending on another nonkey attribute in 3NF.

To normalize the relation completely we need to make the relation go from 1NF to 2NF and then over to 3NF. To make the relation go from 1NF to 2NF split the relation into two different relations. The first could be CAR_SALE1(Car, Date_sold, Discount) and the second could be CAR_SALE2(CAR, Salesperson, Commission).
To then make the relation go from 2NF to 3NF we need to split CAR_SALE2 into two new relations. The first split from CAR_SALE2 can be CAR_SALE3(Car, Salesperson) and the second part of the split can be COM(Salesperson, Commission).
With the new relations the attributes are now all fully dependent on the relations primary key and there exists no transitive dependency in the relation, thus the relation is in 3NF.

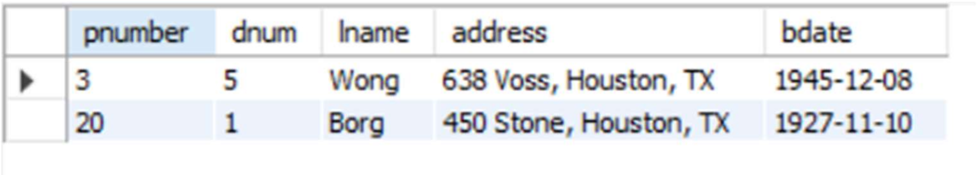
Task 3

a)
```
SELECT e.fname, e.lname
From employee e
JOIN works_on w ON e.ssn = w.essn
JOIN project p ON w.pno = p.pnumber
WHERE p.pname = ‘Computerization’;
```



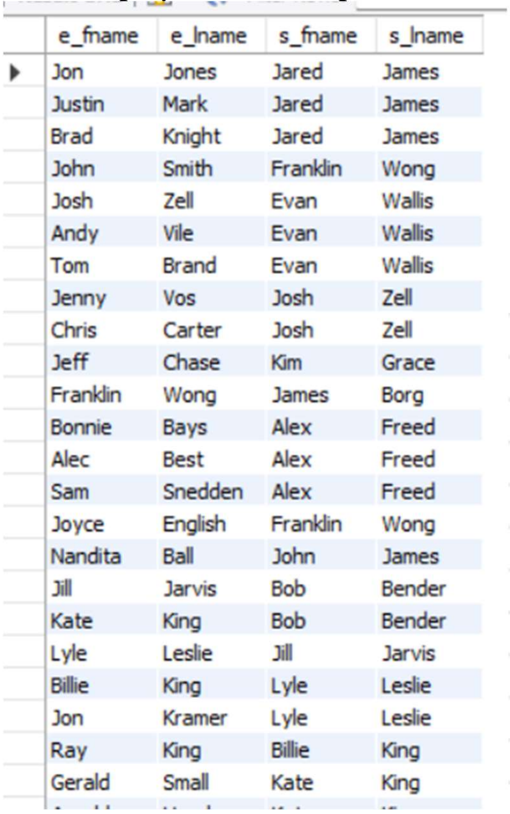
fname	lname
Franklin	Wong
Ahmad	Jabbar
Alicia	Zelaya

b)
```
SELECT p.pnumber, p.dnum, e.lname, e.address, e.bdate
FROM project p
JOIN department d ON p.dnum = d.dnumber
JOIN employee e ON d.mgrssn = e.ssn
WHERE p.plocation = 'Houston';
```

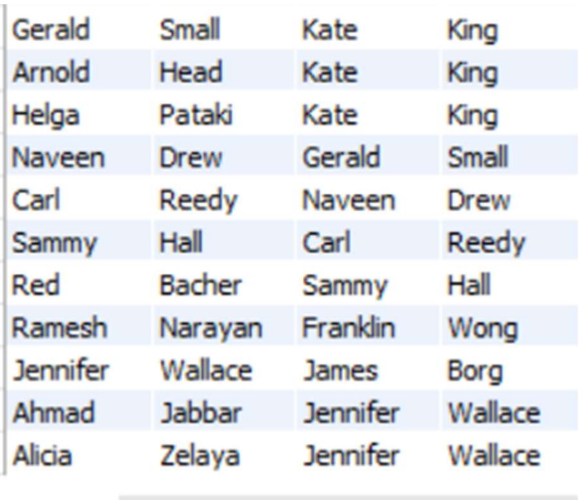


	pnumber	dnum	lname	address	bdate
▶	3	5	Wong	638 Voss, Houston, TX	1945-12-08
	20	1	Borg	450 Stone, Houston, TX	1927-11-10

c) SELECT
e.fname AS e_fname,
e.lname AS e_lname,
s.fname AS s_fname,
s.lname AS s_lname
FROM employee e
JOIN employee s ON e.superssn = s.ssn;

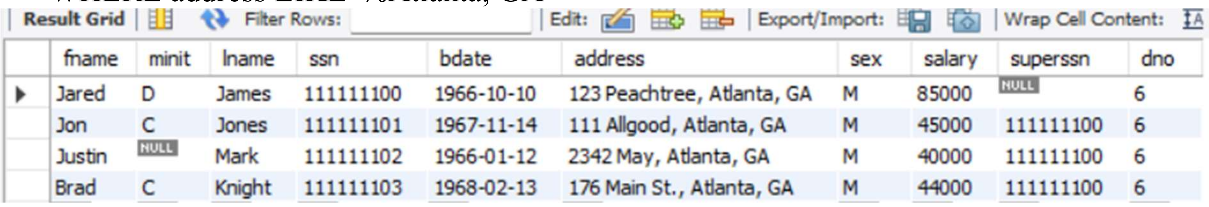


	e_fname	e_lname	s_fname	s_lname
▶	Jon	Jones	Jared	James
	Justin	Mark	Jared	James
	Brad	Knight	Jared	James
	John	Smith	Franklin	Wong
	Josh	Zell	Evan	Wallis
	Andy	Vile	Evan	Wallis
	Tom	Brand	Evan	Wallis
	Jenny	Vos	Josh	Zell
	Chris	Carter	Josh	Zell
	Jeff	Chase	Kim	Grace
	Franklin	Wong	James	Borg
	Bonnie	Bays	Alex	Freed
	Alec	Best	Alex	Freed
	Sam	Snedden	Alex	Freed
	Joyce	English	Franklin	Wong
	Nandita	Ball	John	James
	Jill	Jarvis	Bob	Bender
	Kate	King	Bob	Bender
	Lyle	Leslie	Jill	Jarvis
	Billie	King	Lyle	Leslie
	Jon	Kramer	Lyle	Leslie
	Ray	King	Billie	King
	Gerald	Small	Kate	King



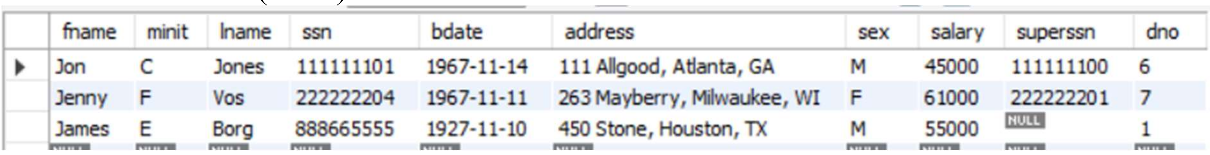
Gerald	Small	Kate	King
Arnold	Head	Kate	King
Helga	Pataki	Kate	King
Naveen	Drew	Gerald	Small
Carl	Reedy	Naveen	Drew
Sammy	Hall	Carl	Reedy
Red	Bacher	Sammy	Hall
Ramesh	Narayan	Franklin	Wong
Jennifer	Wallace	James	Borg
Ahmad	Jabbar	Jennifer	Wallace
Alicia	Zelaya	Jennifer	Wallace

d) SELECT *
FROM employee
WHERE address LIKE '%Atlanta, GA'



	fname	minit	lname	ssn	bdate	address	sex	salary	superssn	dno
▶	Jared	D	James	111111100	1966-10-10	123 Peachtree, Atlanta, GA	M	85000	NULL	6
	Jon	C	Jones	111111101	1967-11-14	111 Allgood, Atlanta, GA	M	45000	111111100	6
	Justin	NULL	Mark	111111102	1966-01-12	2342 May, Atlanta, GA	M	40000	111111100	6
	Brad	C	Knight	111111103	1968-02-13	176 Main St., Atlanta, GA	M	44000	111111100	6

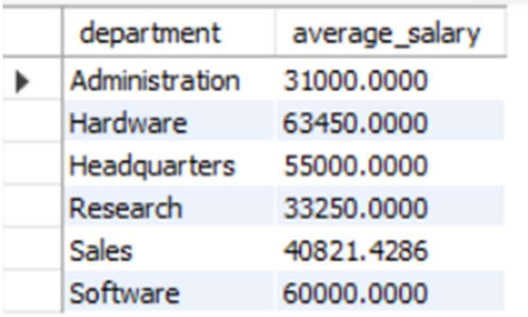
e) SELECT *
FROM employee
WHERE Month(bdate) = 11



	fname	minit	lname	ssn	bdate	address	sex	salary	superssn	dno
▶	Jon	C	Jones	111111101	1967-11-14	111 Allgood, Atlanta, GA	M	45000	111111100	6
	Jenny	F	Vos	222222204	1967-11-11	263 Mayberry, Milwaukee, WI	F	61000	222222201	7
	James	E	Borg	888665555	1927-11-10	450 Stone, Houston, TX	M	55000	NULL	1

f) SELECT d.dname AS department,

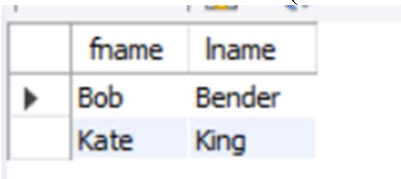
```
AVG(e.salary) AS average_salary
FROM department d
JOIN employee e ON d.dnumber = e.dno
GROUP BY d.dname
```



department	average_salary
Administration	31000.0000
Hardware	63450.0000
Headquarters	55000.0000
Research	33250.0000
Sales	40821.4286
Software	60000.0000

g)

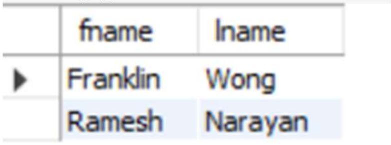
```
SELECT fname, lname
FROM employee
WHERE ssn NOT IN (SELECT essn FROM works_on);
```



fname	lname
Bob	Bender
Kate	King

h)

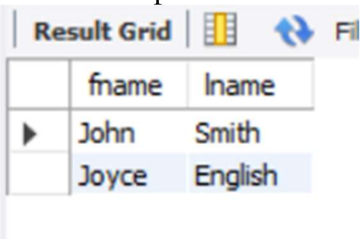
```
SELECT e.fname, e.lname
FROM employee e
JOIN works_on w ON e.ssn = w.essn
JOIN project p ON w.pno = p.pnumber
WHERE e.dno = 5
AND e.salary > 30000
AND p.pname = 'ProductZ';
```



fname	lname
Franklin	Wong
Ramesh	Narayan

i)

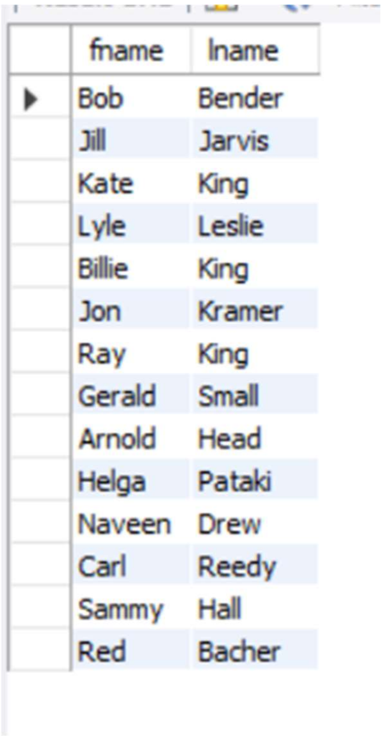
```
SELECT fname, lname
From employee
WHERE address like '%Houston, TX'
AND superssn = 333445555
```



fname	lname
John	Smith
Joyce	English

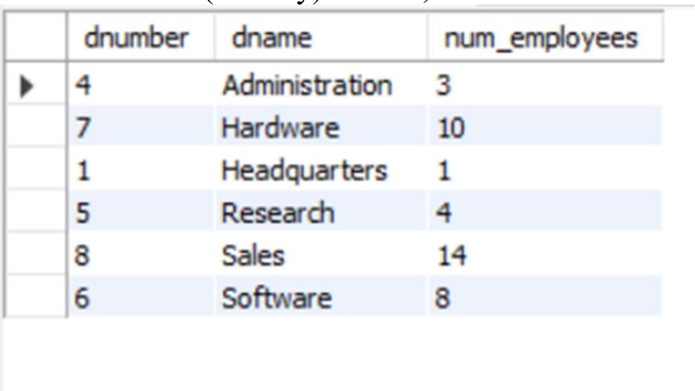
j)

```
SELECT e.fname, e.lname
FROM employee e
WHERE e.dno = (SELECT d.dnumber
FROM department d
JOIN employee e ON d.dnumber = e.dno
GROUP BY d.dnumber
ORDER BY MAX(e.salary) DESC LIMIT 1);
```



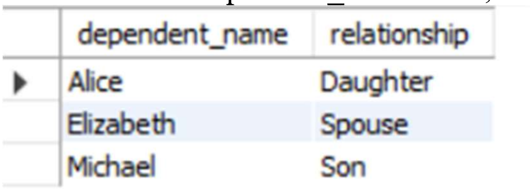
	fname	lname
▶	Bob	Bender
	Jill	Jarvis
	Kate	King
	Lyle	Leslie
	Billie	King
	Jon	Kramer
	Ray	King
	Gerald	Small
	Arnold	Head
	Helga	Pataki
	Naveen	Drew
	Carl	Reedy
	Sammy	Hall
	Red	Bacher

k) SELECT d.dnumber, d.dname, COUNT(*) AS num_employees
FROM department d
JOIN employee e ON d.dnumber = e.dno
GROUP BY d.dnumber, d.dname
HAVING AVG(e.salary)>30000;



	dnumber	dname	num_employees
▶	4	Administration	3
	7	Hardware	10
	1	Headquarters	1
	5	Research	4
	8	Sales	14
	6	Software	8

l) SELECT d.dependent_name, d.relationship
FROM dependent d
JOIN employee e ON d.essn = e.ssn
JOIN employee s ON e.superssn = s.ssn
WHERE s.ssn = ‘333445555’
ORDER BY d.dependent_name ASC;



	dependent_name	relationship
▶	Alice	Daughter
	Elizabeth	Spouse
	Michael	Son