

1DV503/1DT903 Database Technology and Modeling

Jesper Wingren

School of Computer Science, Physics
and Mathematics, Linnaeus University, Sweden

jw223rn@student.lnu.se

Assignment 2

Task 1:

A.

{Manufacturer,serial number} -> Model, Batch, Capacity, Retailer

B.

Model number -> manufacturer

C.

Batch -> Model

Task 2:

The relation is in 1NF because it has a primary key that uniquely identifies each record and it doesn't contain any multivalued or nested relations. But the relation is not in 2NF or 3NF because it contains an attribute that is not dependent on the whole primary key, for example the commission is only dependent on the salesperson. And for the relation to be 3NF it must first be in 2NF and it is not allowed to have a nonkey attribute dependent on another nonkey attribute, but the discount is dependent on the date_sold attribute which leads to it not being 3NF.

To normalize it completely you first need to make it 2NF by splitting it into two different relations, CAR_SALE_1(Car,Date_sold,Discount) and CAR_SALE_2(Car,Salesperson, Commission).

To normalize it into 3NF we split the CAR_SALE_2 into two new relations, CAR_SALE_3(Car,Salesperson) and COMMISSION(Salesperson, commission).

With these relations all attributes are fully dependent on the primary key of each relation and no transitive dependencies exist. Thus, the relation is 3NF.

Task 3:

A.

Query:

SELECT fname,lname

```

FROM EMPLOYEE AS E
INNER JOIN WORKS _ON AS W ON E.ssn = W.essn
INNER JOIN PROJECT AS P ON W.pno = P.pnumber
WHERE P.pname = 'computerization';

```

Output:

	fname	lname
▶	Franklin	Wong
	Ahmad	Jabbar
	Alicia	Zelaya

B.

Query:

```

SELECT pnumber,dnumber,lname,address,bdate
FROM PROJECT AS P
INNER JOIN DEPARTMENT AS D ON P.dnum = D.dnumber
INNER JOIN EMPLOYEE AS E ON D.mgrssn = E.ssn
WHERE plocation='Houston';

```

Output:

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

C.

Query:

```

SELECT E.fname,E.lname,S.fname as supervisor_fname,S.lname as
supervisor_lname
FROM EMPLOYEE AS E
LEFT JOIN EMPLOYEE AS S ON E.superssn = S.ssn;

```

Output:

	fname	lname	supervisor_fname	supervisor_lname
►	Jared	James	NULL	NULL
	Jon	Jones	Jared	James
	Justin	Mark	Jared	James
	Brad	Knight	Jared	James
	John	Smith	Franklin	Wong
	Evan	Wallis	NULL	NULL
	Josh	Zell	Evan	Wallis
	Andy	Vile	Evan	Wallis
	Tom	Brand	Evan	Wallis
	Jenny	Vos	Josh	Zell
	Chris	Carter	Josh	Zell
	Kim	Grace	NULL	NULL
	Jeff	Chase	Kim	Grace
	Frank...	Wong	James	Borg
	Alex	Freed	NULL	NULL
	Bonnie	Bays	Alex	Freed
	Alec	Best	Alex	Freed
	Sam	Sned...	Alex	Freed
	Joyce	English	Franklin	Wong
	.	.	NULL	NULL
	John	James	NULL	NULL
	Nandita	Ball	John	James
	Bob	Bender	NULL	NULL
	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
	Arnold	Head	Kate	King
	Helga	Pataki	Kate	King
	Naveen	Drew	Gerald	Small
	Carl	Reedy	Naveen	Drew
	Sam...	Hall	Carl	Reedy
	Red	Bacher	Sammy	Hall
	Ram...	Naray...	Franklin	Wong
	James	Borg	NULL	NULL
	Jennif...	Wallace	James	Borg
	Ahmad	Jabbar	Jennifer	Wallace
	Alicia	Zelaya	Jennifer	Wallace

D.

Query:

```
SELECT *  
FROM EMPLOYEE  
WHERE address LIKE '%Atlanta, GA';
```

Output:

	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.

Query:

```
SELECT *  
FROM EMPLOYEE  
WHERE MONTH(bdate)='11'
```

Output:

	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.

Query:

```
SELECT dname, AVG(E.salary) as average_salary  
FROM EMPLOYEE AS E  
INNER JOIN DEPARTMENT AS D ON E.dno = D.dnumber  
GROUP BY D.dname;
```

Output:

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

G.

Query:

```

SELECT fname,lname
FROM EMPLOYEE AS E
LEFT JOIN WORKS_ON AS W ON E.ssn = W.essn
WHERE W.essn IS NULL;

```

Output:

	fname	lname
▶	Bob	Bender
	Kate	King

H.

Query:

```

SELECT fname,lname
FROM EMPLOYEE AS E
INNER JOIN WORKS_ON AS W ON E.ssn = W.essn
INNER JOIN PROJECT AS P ON W.pno = P.pnumber
INNER JOIN DEPARTMENT AS D ON E.dno = D.dnumber
WHERE D.dnumber = '5' AND E.salary > '30000' AND P.pname = 'ProductZ';

```

Output:

	fname	lname
▶	Franklin	Wong
	Ramesh	Narayan

I.

Query:

```

SELECT fname,lname
FROM EMPLOYEE AS E
INNER JOIN DEPARTMENT AS D ON E.dno = D.dnumber
WHERE E.address LIKE '%Houston, TX%' AND D.mgrssn = '333445555';

```

Output:

	fname	lname
▶	John	Smith
	Franklin	Wong
	Joyce	English

J.

Query:

```

SELECT fname,lname
FROM EMPLOYEE AS E
INNER JOIN DEPARTMENT AS D ON E.dno=D.dnumber
WHERE D.dnumber = (
SELECT dno
  FROM EMPLOYEE
 GROUP BY dno
 ORDER BY MAX(salary) DESC
 LIMIT 1
);

```

Output:

	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.

Query:

```

SELECT dnumber,dname,COUNT(ssn) as num_employees
FROM EMPLOYEE AS E
INNER JOIN DEPARTMENT AS D ON E.dno = D.dnumber
GROUP BY D.dnumber,D.dname
HAVING AVG(E.salary) > '30000';

```

Output:

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

L.

Query:

```
SELECT dependent_name,relationship
FROM DEPENDENT AS DE
INNER JOIN EMPLOYEE AS E ON DE.essn = E.ssn
WHERE E.superssn = '333445555'
ORDER BY DE.dependent_name ASC;
```

Output:

	dependent_name	relationship
►	Alice	Daughter
	Elizabeth	Spouse
	Michael	Son