DATA MANAGEMENT FOR DATA SCIENCE

HOMEWORK 1 – 2

TEACHERS: PROF. LEMBO DOMENICO

PROF. ROSATI RICCARDO

PROVIDERS: MELIKA SADAT PARPINCHI 1880156

MOUSAALREZA DASTMARD 1852433

SUMMARY:

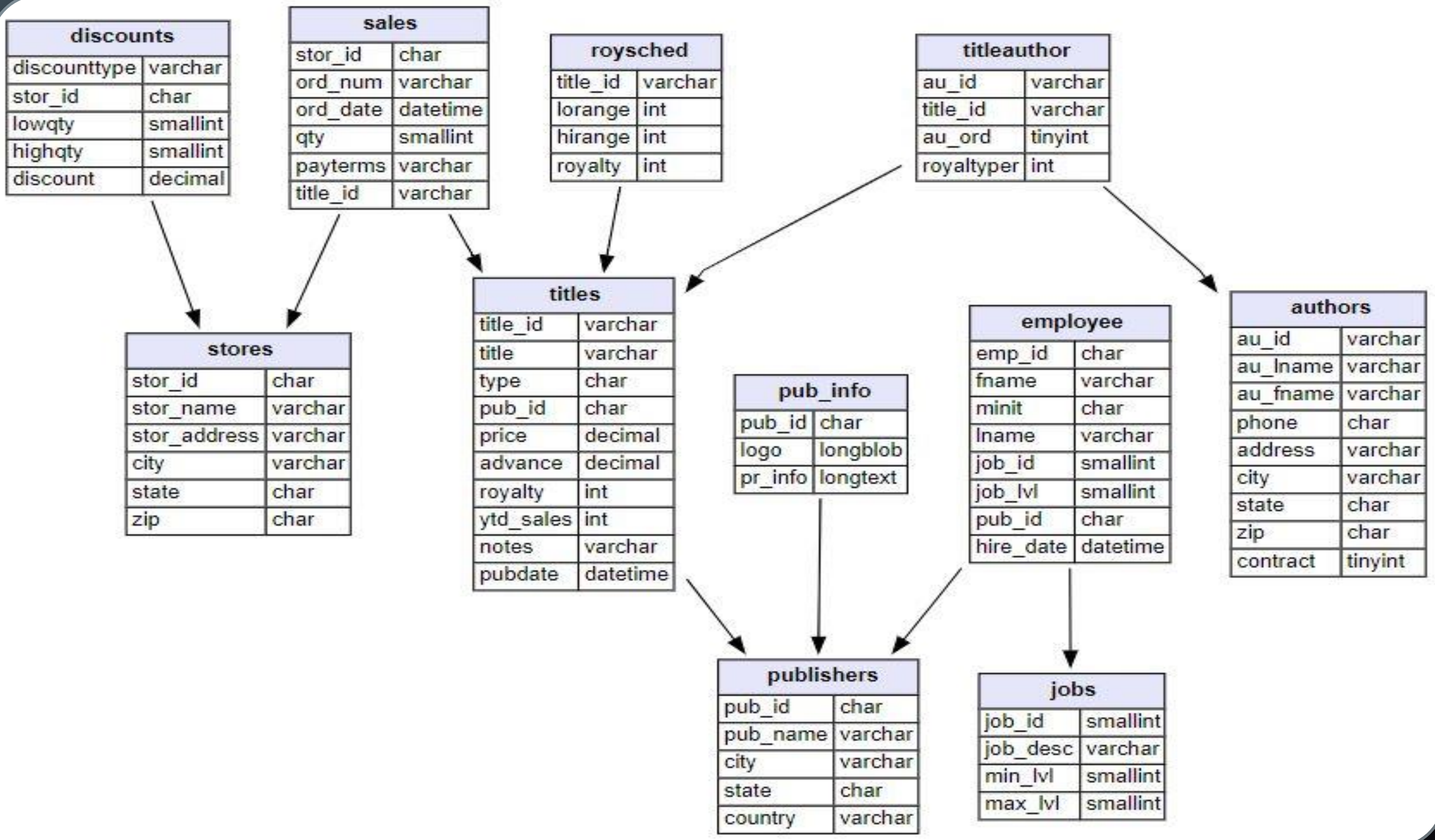
DATASET: PUBS, 11 TABLES

# OF QUERIES: 20 (INCLUDING DUPLICATES)

# OF UNIQUE QUERIES: 11

PUBS DATASET

HTTPS://RELATIONAL.FIT.CVUT.CZ/DATASET/PUBS



discounts	
discounttype	varchar
stor_id	char
lowqty	smallint
highqty	smallint
discount	decimal

sales	
stor_id	char
ord_num	varchar
ord_date	datetime
qty	smallint
payterms	varchar
title_id	varchar

roysched	
title_id	varchar
lorange	int
hirange	int
royalty	int

titleauthor	
au_id	varchar
title_id	varchar
au_ord	tinyint
royaltyper	int

stores	
stor_id	char
stor_name	varchar
stor_address	varchar
city	varchar
state	char
zip	char

titles	
title_id	varchar
title	varchar
type	char
pub_id	char
price	decimal
advance	decimal
royalty	int
ytd_sales	int
notes	varchar
pubdate	datetime

pub_info	
pub_id	char
logo	longblob
pr_info	longtext

employee	
emp_id	char
fname	varchar
minit	char
lname	varchar
job_id	smallint
job_lvl	smallint
pub_id	char
hire_date	datetime

authors	
au_id	varchar
au_lname	varchar
au_fname	varchar
phone	char
address	varchar
city	varchar
state	char
zip	char
contract	tinyint

publishers	
pub_id	char
pub_name	varchar
city	varchar
state	char
country	varchar

jobs	
job_id	smallint
job_desc	varchar
min_lvl	smallint
max_lvl	smallint

DATASET DETAILS:

Data types:

- Numeric
- String
- Datetime

Size:

- 400 KB

Count of tables:

- 11

Count of rows:

- 255

Count of columns:

- 64

Missing values:

- Yes

Target table:

- Titles

Target column:

- Ytd_sales

Target_ID:

- Title_ID

Target timestamp:

- pubdate

TABLE DETAILS:

	Tables:	**Primary Keys:**
i.	Titles	Title_id
ii.	Authors	Au_id
iii.	Titleauthor	Au_id + title_id
iv.	Sales	Title_id + stor_id + ord_num
v.	Stores	Stor_id
vi.	discounts	Stor_id
vii.	Publishers	Pub_id
viii.	Pub_info	Pub_id
ix.	Employee	Emp_id
x.	Jobs	Job_id
xi.	Roysched	Title_id

HW1: nested queries

HW2: rewriting, indexing, adding views

Q1-1, Q2-1-1, Q2-1-2: LIST OF PUBLISHERS THAT DON'T HAVE BUSINESS BOOK

✓ Using **“not exists”:**

```
select * from publishers where not exists
        (select * from titles where titles.pub_id
= publishers.pub_id and type = 'business');
```

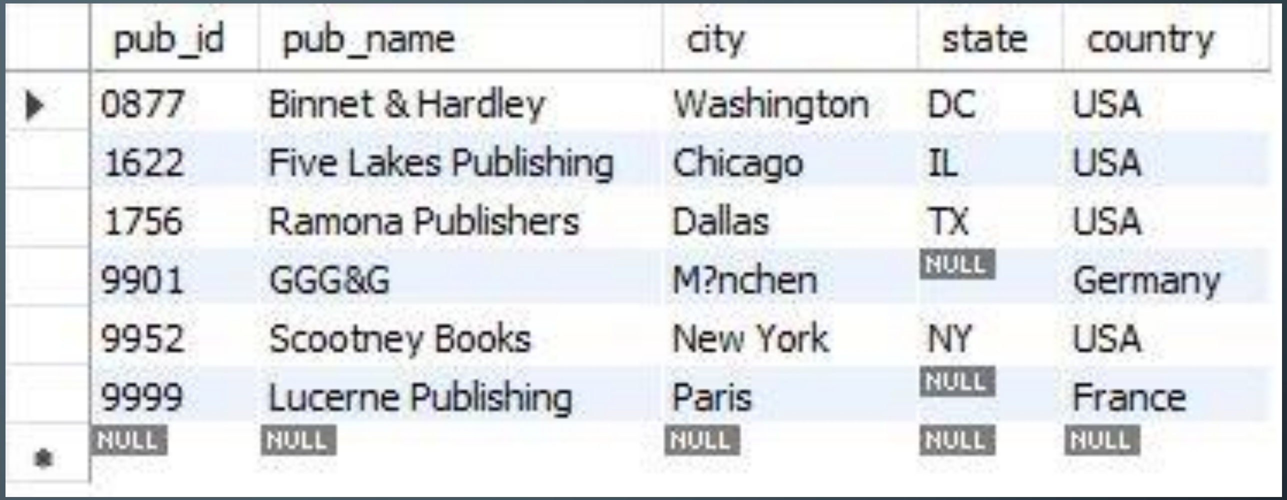
Optimizing (Using View and index):

```
create index i_type on titles(type);
create view business_pub_ids as select
pub_id from titles where type =
'business';
```

✓ Using **“not in”:**

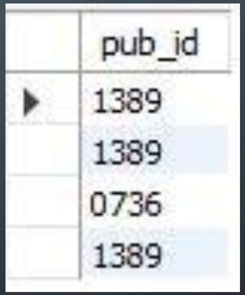
```
select * from publishers where pub_id not in
        (select * from business_pub_ids );
```

CODE RESULT:



	pub_id	pub_name	city	state	country
▶	0877	Binnet & Hardley	Washington	DC	USA
	1622	Five Lakes Publishing	Chicago	IL	USA
	1756	Ramona Publishers	Dallas	TX	USA
	9901	GGG&G	M?nchen	NULL	Germany
	9952	Scootney Books	New York	NY	USA
	9999	Lucerne Publishing	Paris	NULL	France
*	NULL	NULL	NULL	NULL	NULL

VIEW RESULT:



	pub_id
▶	1389
	1389
	0736
	1389

Our analyze:

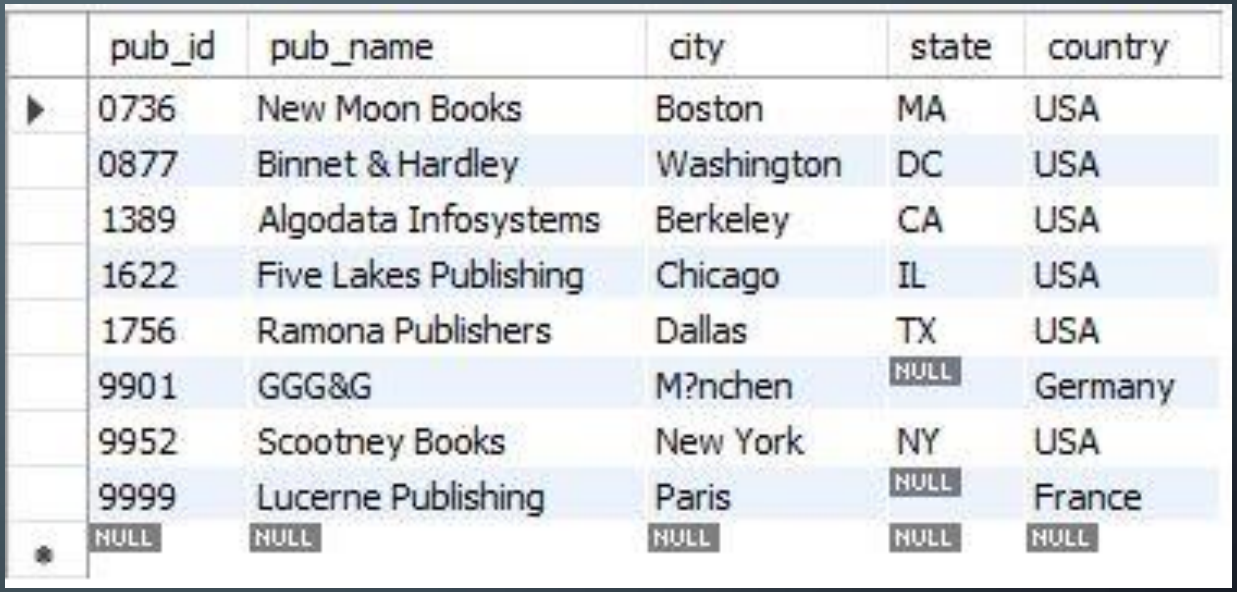
There 4 publishers located in the USA and two in Germany and France.

Q1-2: LIST OF PUBLISHERS THAT HAVE PUBLISHED BOOKS THAT HAVE MOD IN THEIR TYPE

✓ Using **“exists”, “Like”:**

```
select * from publishers where  exists
        (select * from titles where type
like '%mod%');
```

CODE RESULT:



	pub_id	pub_name	city	state	country
▶	0736	New Moon Books	Boston	MA	USA
	0877	Binnet & Hardley	Washington	DC	USA
	1389	Algodata Infosystems	Berkeley	CA	USA
	1622	Five Lakes Publishing	Chicago	IL	USA
	1756	Ramona Publishers	Dallas	TX	USA
	9901	GGG&G	M?nchen	NULL	Germany
	9952	Scootney Books	New York	NY	USA
	9999	Lucerne Publishing	Paris	NULL	France
*	NULL	NULL	NULL	NULL	NULL

Our analyze:

Mostly the publishers have books of type %mod% are located in the USA

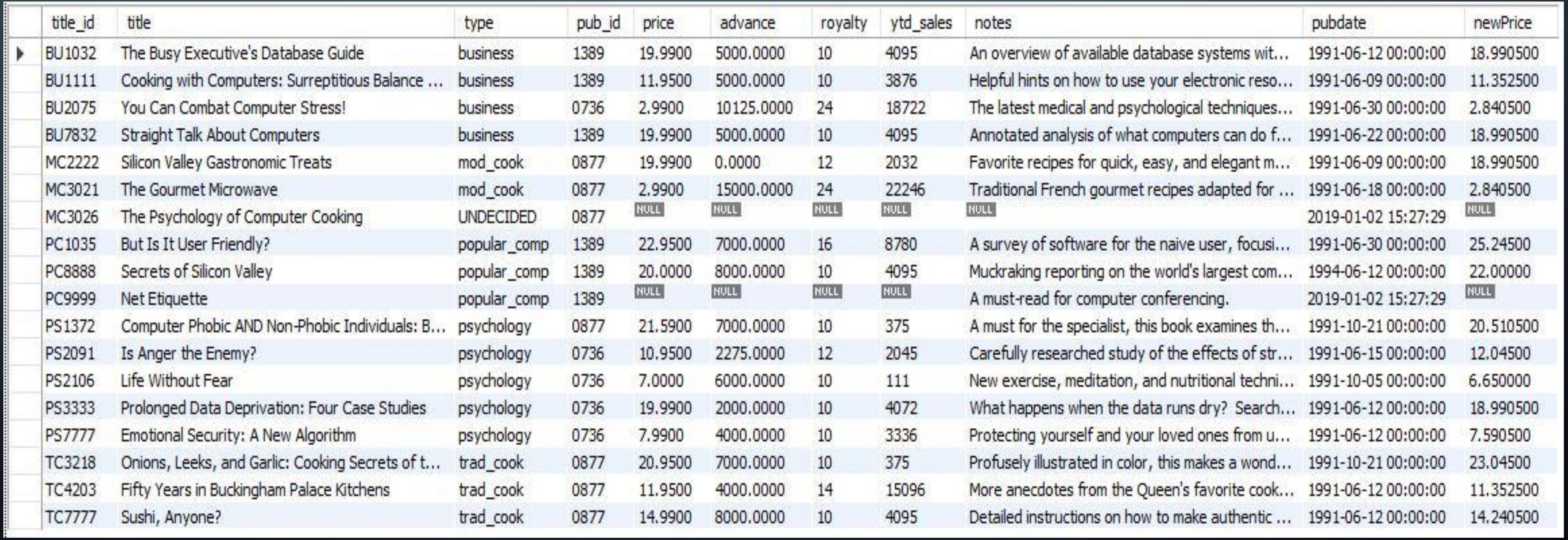
Q1-3: RAISING THE PRICE BY 10% FOR THOSE BOOKS HAVE TOTAL SALE MORE THAN 500 ELSE DECREASING BY 5%

Our analyze:

Comparing the columns price and newPrice we can see that mostly the new calculated price is less than previous price.

✓ Using **“case when”**, “group by”, “having” :

```
select * ,
case when
        title_id in (select titles.title_id from titles inner join
        sales on sales.title_id = titles.title_id group by
        titles.title_id having sum(qty*price) > 500)
then price * 1.1
else price * .95
end as newPricefrom titles ;
```



	title_id	title	type	pub_id	price	advance	royalty	ytd_sales	notes	pubdate	newPrice
▶	BU1032	The Busy Executive's Database Guide	business	1389	19.9900	5000.0000	10	4095	An overview of available database systems wit...	1991-06-12 00:00:00	18.990500
	BU1111	Cooking with Computers: Surreptitious Balance ...	business	1389	11.9500	5000.0000	10	3876	Helpful hints on how to use your electronic reso...	1991-06-09 00:00:00	11.352500
	BU2075	You Can Combat Computer Stress!	business	0736	2.9900	10125.0000	24	18722	The latest medical and psychological techniques...	1991-06-30 00:00:00	2.840500
	BU7832	Straight Talk About Computers	business	1389	19.9900	5000.0000	10	4095	Annotated analysis of what computers can do f...	1991-06-22 00:00:00	18.990500
	MC2222	Silicon Valley Gastronomic Treats	mod_cook	0877	19.9900	0.0000	12	2032	Favorite recipes for quick, easy, and elegant m...	1991-06-09 00:00:00	18.990500
	MC3021	The Gourmet Microwave	mod_cook	0877	2.9900	15000.0000	24	22246	Traditional French gourmet recipes adapted for ...	1991-06-18 00:00:00	2.840500
	MC3026	The Psychology of Computer Cooking	UNDECIDED	0877	NULL	NULL	NULL	NULL	NULL	2019-01-02 15:27:29	NULL
	PC1035	But Is It User Friendly?	popular_comp	1389	22.9500	7000.0000	16	8780	A survey of software for the naive user, focusi...	1991-06-30 00:00:00	25.24500
	PC8888	Secrets of Silicon Valley	popular_comp	1389	20.0000	8000.0000	10	4095	Muckraking reporting on the world's largest com...	1994-06-12 00:00:00	22.00000
	PC9999	Net Etiquette	popular_comp	1389	NULL	NULL	NULL	NULL	A must-read for computer conferencing.	2019-01-02 15:27:29	NULL
	PS1372	Computer Phobic AND Non-Phobic Individuals: B...	psychology	0877	21.5900	7000.0000	10	375	A must for the specialist, this book examines th...	1991-10-21 00:00:00	20.510500
	PS2091	Is Anger the Enemy?	psychology	0736	10.9500	2275.0000	12	2045	Carefully researched study of the effects of str...	1991-06-15 00:00:00	12.04500
	PS2106	Life Without Fear	psychology	0736	7.0000	6000.0000	10	111	New exercise, meditation, and nutritional techni...	1991-10-05 00:00:00	6.650000
	PS3333	Prolonged Data Deprivation: Four Case Studies	psychology	0736	19.9900	2000.0000	10	4072	What happens when the data runs dry? Search...	1991-06-12 00:00:00	18.990500
	PS7777	Emotional Security: A New Algorithm	psychology	0736	7.9900	4000.0000	10	3336	Protecting yourself and your loved ones from u...	1991-06-12 00:00:00	7.590500
	TC3218	Onions, Leeks, and Garlic: Cooking Secrets of t...	trad_cook	0877	20.9500	7000.0000	10	375	Profusely illustrated in color, this makes a wond...	1991-10-21 00:00:00	23.04500
	TC4203	Fifty Years in Buckingham Palace Kitchens	trad_cook	0877	11.9500	4000.0000	14	15096	More anecdotes from the Queen's favorite cook...	1991-06-12 00:00:00	11.352500
	TC7777	Sushi, Anyone?	trad_cook	0877	14.9900	8000.0000	10	4095	Detailed instructions on how to make authentic ...	1991-06-12 00:00:00	14.240500

HW1: joins, aggregations, nested queries

HW2: tables derived from the existing database tables

Q1-4: TAX CALCULATION FOR EACH BOOK BASED ON TOTAL SALE IF TOTAL SALE IS LESS THAN 200 THEN TAX = 0 IF TOTAL SALE IS LESS THAN 500 THEN TAX = (TOTAL SALE - 200)*5% IF TOTAL SALE IS LESS THAN 800 THEN TAX = 15 + (TOTAL SALE - 500)*10% IF TOTAL SALE IS LESS THAN 1000 THEN TAX = 45 + (TOTAL SALE - 800)*15% ELSE TAX = 75 + (TOTAL SALE - 1000)*20%

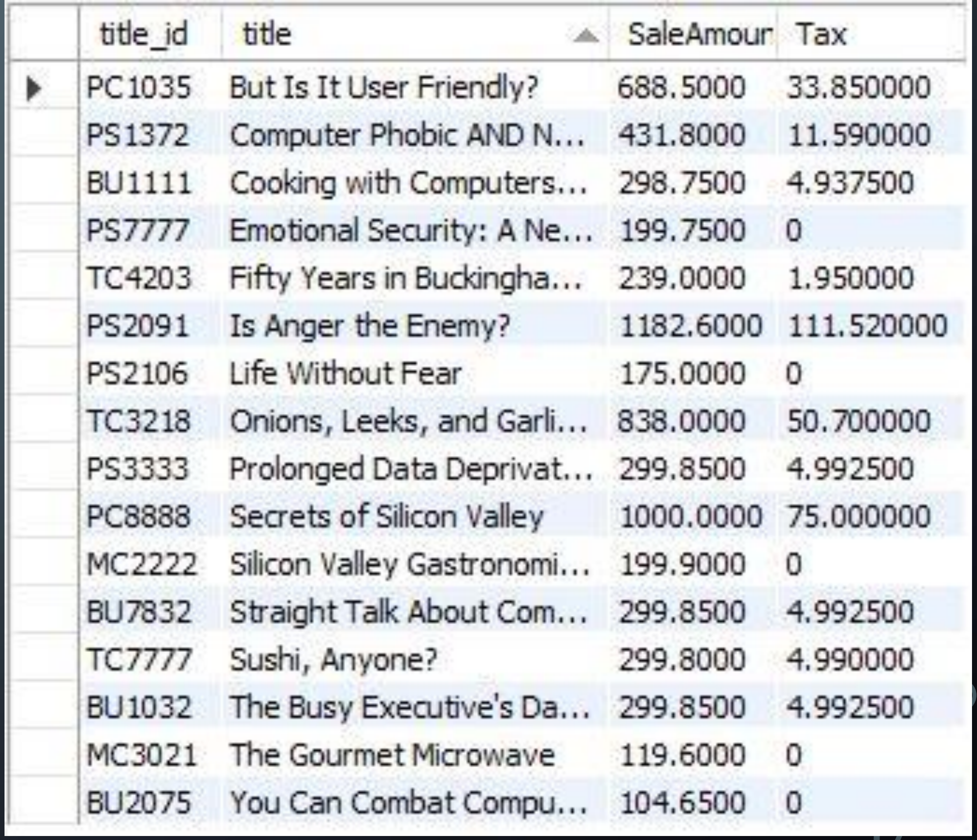
✓ Using “case when”, “drived query”, “group by”

```
select * ,              case when SaleAmount < 200     then 0
                             when SaleAmount < 500     then 0+(SaleAmount - 200)  * .05

                             when SaleAmount < 800     then 0 + 15 +(SaleAmount - 500)  * .10
                             when SaleAmount < 1000    then 0 + 15 + 30 +(SaleAmount - 800)  * .15

                        else                            0 + 15 + 30 + 30 + (SaleAmount - 1000) * .20
 end as Tax
from (select titles.title_id , title , sum(qty*price) as SaleAmount from sales inner join titles
on titles.title_id = sales.title_id
group by titles.title_id , title) as d ;
```

CODE RESULT:



title_id	title	SaleAmour	Tax
PC1035	But Is It User Friendly?	688.5000	33.850000
PS1372	Computer Phobic AND N...	431.8000	11.590000
BU1111	Cooking with Computers...	298.7500	4.937500
PS7777	Emotional Security: A Ne...	199.7500	0
TC4203	Fifty Years in Buckingha...	239.0000	1.950000
PS2091	Is Anger the Enemy?	1182.6000	111.520000
PS2106	Life Without Fear	175.0000	0
TC3218	Onions, Leeks, and Garli...	838.0000	50.700000
PS3333	Prolonged Data Deprivat...	299.8500	4.992500
PC8888	Secrets of Silicon Valley	1000.0000	75.000000
MC2222	Silicon Valley Gastronomi...	199.9000	0
BU7832	Straight Talk About Com...	299.8500	4.992500
TC7777	Sushi, Anyone?	299.8000	4.990000
BU1032	The Busy Executive's Da...	299.8500	4.992500
MC3021	The Gourmet Microwave	119.6000	0
BU2075	You Can Combat Compu...	104.6500	0

Our analyze:

Rarely we can find publishers that have to pay TAX more than 100$ based on TAX scenario defined above. And there exist publishers have not to pay TAX.

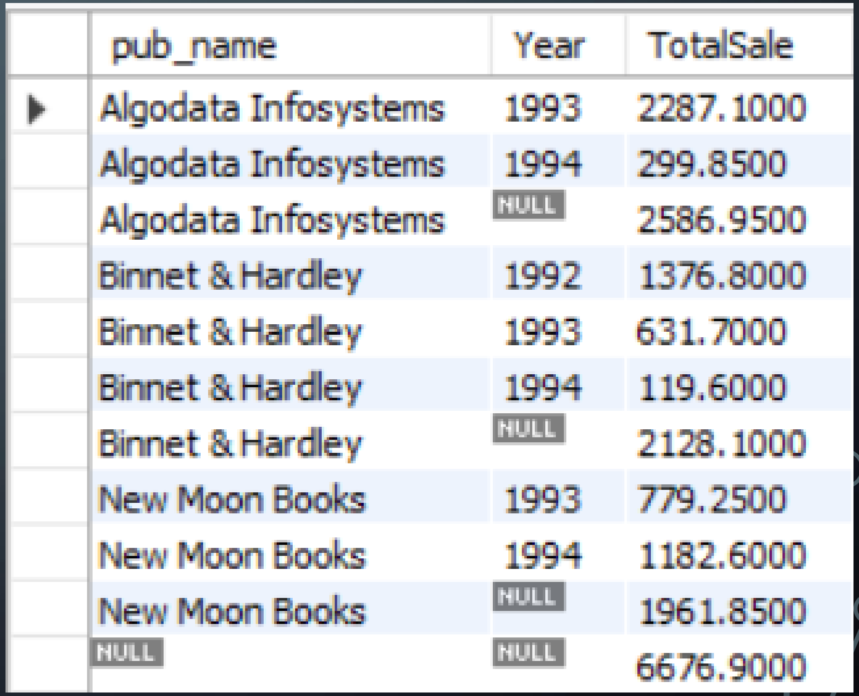
Q1-5: TOTAL SALE OF PUBLISHERS IN DIFFERENT YEARS AND IN OVERALL.

✓ Using **“group by”, “rollup”, “YEAR”, “sum”:**

```
select  pub_name  , YEAR(ord_date) as Year , sum(qty * price ) as TotalSale
from sales inner join
        titles on titles.title_id = sales.title_id inner join
        publishers on publishers.pub_id = titles.pub_id
group by  pub_name , YEAR(ord_date)
with rollup;
```

The ROLLUP generates the subtotal row every time the product line changes and the grand total at the end of the result.

CODE RESULT:



	pub_name	Year	TotalSale
▶	Algodata Infosystems	1993	2287.1000
	Algodata Infosystems	1994	299.8500
	Algodata Infosystems	NULL	2586.9500
	Binnet & Hardley	1992	1376.8000
	Binnet & Hardley	1993	631.7000
	Binnet & Hardley	1994	119.6000
	Binnet & Hardley	NULL	2128.1000
	New Moon Books	1993	779.2500
	New Moon Books	1994	1182.6000
	New Moon Books	NULL	1961.8500
	NULL	NULL	6676.9000

Our analyze:

There are only 3 publishers that have sold books listed in titles table The results shows that each publishers almost sold same amount of books And the total sold per year is decreasing

HW1: queries with negated subqueries, joins

HW2: rewriting queries

Q1-6-1, Q1-6-2: LIST OF AUTHORS THAT DON'T HAVE BOOKS

✓ Using **"is null"** :

```
select *
from authors
left join titleauthor on titleauthor.au_id =
authors.au_id
where title_id is null;
```

Optimizing (Subquery instead of join):

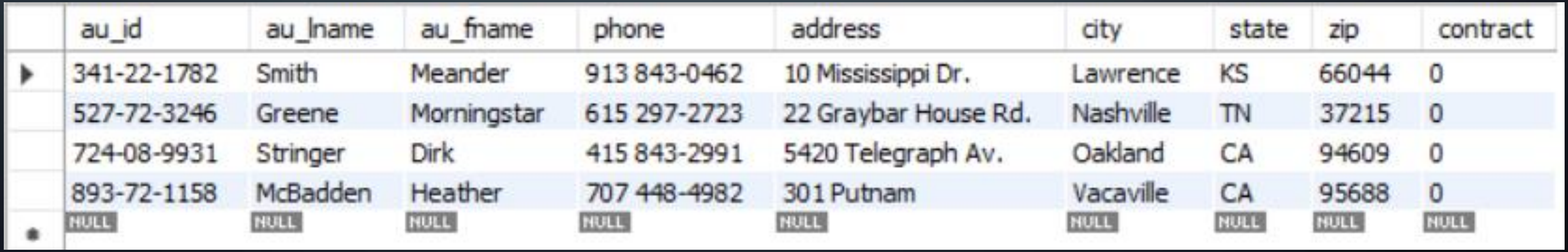
✓ Using "not in" :

```
select *
from authors
where au_id not in
        (select au_id
        from titleauthor);
```

Our analyze:

There are 4 authors that haven't published any book yet

CODE RESULT:



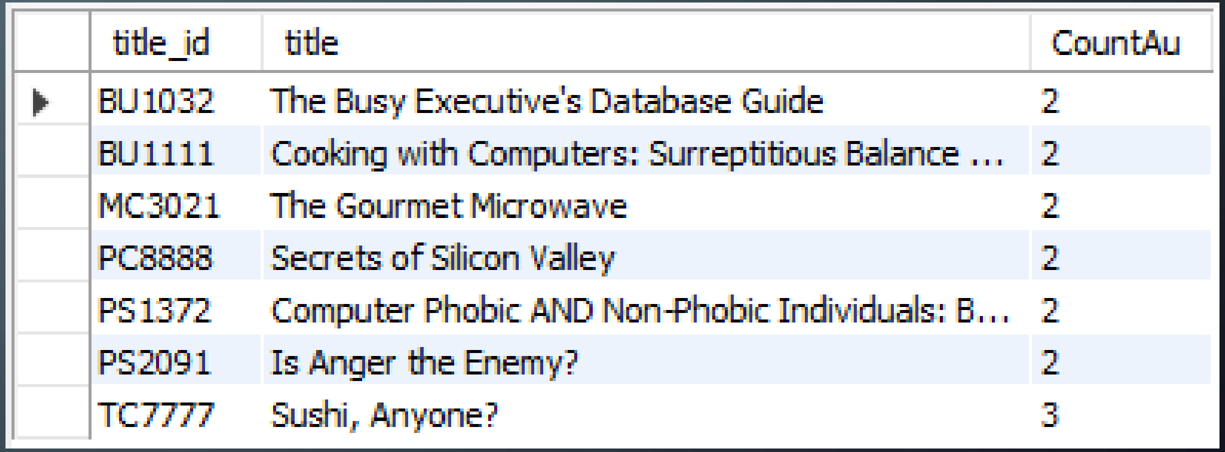
	au_id	au_lname	au_fname	phone	address	city	state	zip	contract
▶	341-22-1782	Smith	Meander	913 843-0462	10 Mississippi Dr.	Lawrence	KS	66044	0
	527-72-3246	Greene	Morningstar	615 297-2723	22 Graybar House Rd.	Nashville	TN	37215	0
	724-08-9931	Stringer	Dirk	415 843-2991	5420 Telegraph Av.	Oakland	CA	94609	0
	893-72-1158	McBadden	Heather	707 448-4982	301 Putnam	Vacaville	CA	95688	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Q1-7: LIST OF BOOKS THAT HAVE AT LEAST 2 AUTHORS IN ASCEND ORDER

- ✓ Using **"Count", "group by", "having", "order by":**

```
select titles.title_id , title , Count(au_id) as CountAu
from titles inner join
          titleauthor  on titleauthor.title_id = titles.title_id
group by titles.title_id , title
having  Count(au_id) > 1
order by 3;
```

CODE RESULT:



	title_id	title	CountAu
▶	BU1032	The Busy Executive's Database Guide	2
	BU1111	Cooking with Computers: Surreptitious Balance ...	2
	MC3021	The Gourmet Microwave	2
	PC8888	Secrets of Silicon Valley	2
	PS1372	Computer Phobic AND Non-Phobic Individuals: B...	2
	PS2091	Is Anger the Enemy?	2
	TC7777	Sushi, Anyone?	3

Our analyze:

Among those books have at least two co-authors, there is only one book with 3 authors and the remain books have only two co-authors

HW1: joins, aggregations, nested queries

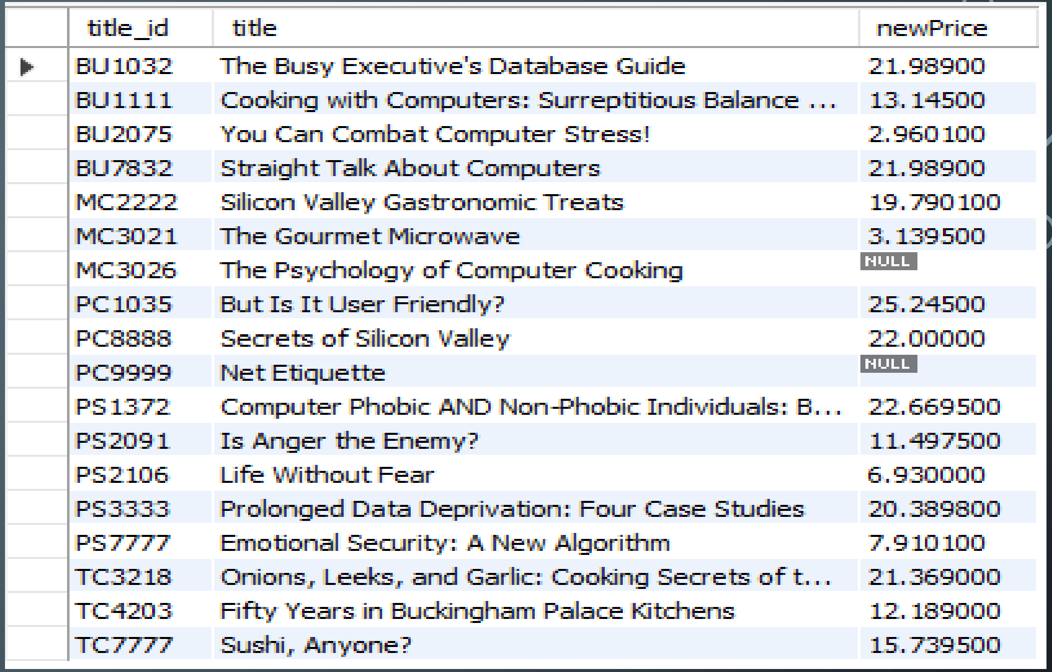
HW2: rewriting queries, adding views

Q1-9, Q2-4-1, Q2-4-2: PRICING BOOK BASED OF VARIOUS CONDITIONS: IF PUBLISHER LOCATED IN CALIFORNIA THEN INCREASE PRICE BY 10%IF THE BOOK HAS MORE THAN 1 AUTHORS THEN INCREASE PRICE BY 5%,IF THE BOOK IS SOLD MORE THAN 200$ THEN INCREASE PRICE BY 2%,ELSE DECREASE PRICE BY 1%

✓ Using **“where”, “group by”, “sum”:**

```
select title_id , title  ,
case when pub_id in
(select pub_id from publishers where state = 'CA')                  then price * 1.1
else case when title_id in
(select title_id from titleauthor group by title_id having count(*) > 1)    then price * 1.05
else case when title_id in
(select titles.title_id from titles inner join
                        sales on sales.title_id = titles.title_id
                        group by titles.title_id
                        having sum(qty*price) > 200)
                                                                    then price * 1.02
```

CODE RESULT:



	title_id	title	newPrice
▶	BU1032	The Busy Executive's Database Guide	21.98900
	BU1111	Cooking with Computers: Surreptitious Balance ...	13.14500
	BU2075	You Can Combat Computer Stress!	2.960100
	BU7832	Straight Talk About Computers	21.98900
	MC2222	Silicon Valley Gastronomic Treats	19.790100
	MC3021	The Gourmet Microwave	3.139500
	MC3026	The Psychology of Computer Cooking	NULL
	PC1035	But Is It User Friendly?	25.24500
	PC8888	Secrets of Silicon Valley	22.00000
	PC9999	Net Etiquette	NULL
	PS1372	Computer Phobic AND Non-Phobic Individuals: B...	22.669500
	PS2091	Is Anger the Enemy?	11.497500
	PS2106	Life Without Fear	6.930000
	PS3333	Prolonged Data Deprivation: Four Case Studies	20.389800
	PS7777	Emotional Security: A New Algorithm	7.910100
	TC3218	Onions, Leeks, and Garlic: Cooking Secrets of t...	21.369000
	TC4203	Fifty Years in Buckingham Palace Kitchens	12.189000
	TC7777	Sushi, Anyone?	15.739500

Next page

Optimizing (Using view):

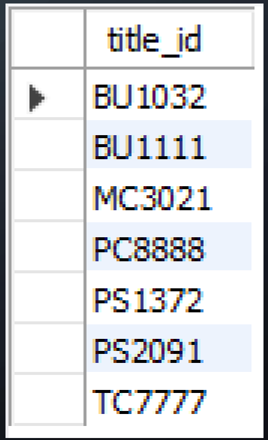
```
create view titleauthor_view as
        select title_id from titleauthor
        group by title_id
        having count(*) > 1;
create view title_view as
select titles.title_id
        from titles inner join
        sales on sales.title_id = titles.title_id
                        group by titles.title_id
                        having sum(qty*price) > 200;
```

✓ Using **“where”:**

```
select title_id , title  ,
        case when pub_id in
        (select pub_id from publishers where state = 'CA')
                                                                then price * 1.1
        else case when title_id in
        (select * from titleauthor_view)                         then price * 1.05
        else case when title_id in
        (select * from title_view)                              then price * 1.02

        else                                                          price * .99
        end end end as newPrice
from titles ;
```

VIEW RESULT:



	title_id
▶	BU1032
	BU1111
	MC3021
	PC8888
	PS1372
	PS2091
	TC7777

Q2-6: MODIFYING THE SCHEMA DATABASE, ADDING INTEGRITY CONSTRAINTS

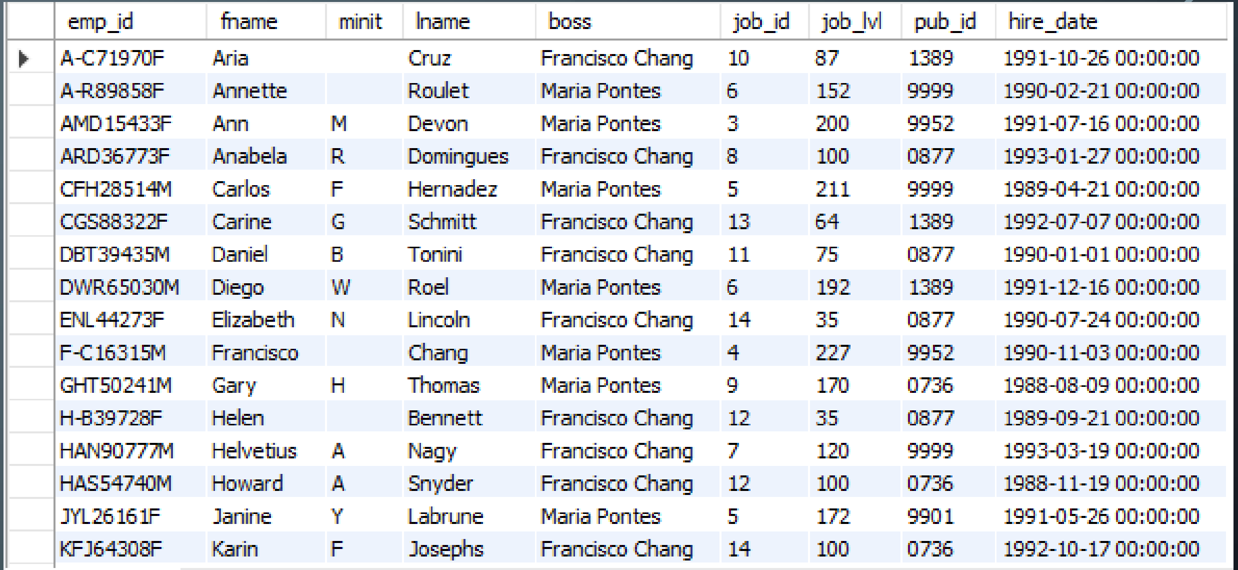
Optimizing (Using View and index):

```
create view boss_view as
        select emp_id, fname, minit, lname,
        case when job_lvl > 150                then 'Maria Pontes'
        else                                   'Francisco Chang'
        end as boss,
  job_id, job_lvl, pub_id, hire_date
  from employee;
```

Modifying table employee and using check constraint:

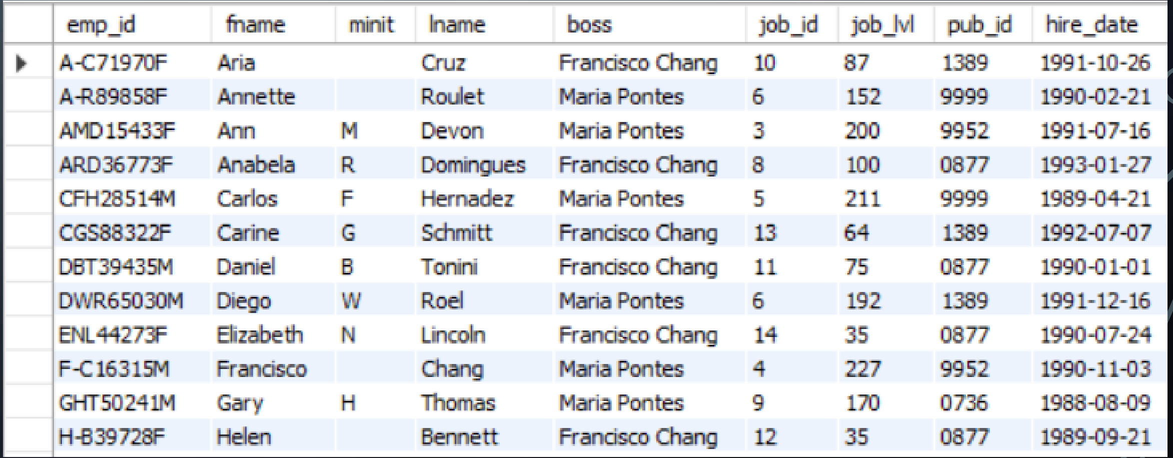
```
alter table employee
ADD  boss varchar(50) check (boss in ('Francisco Chang', 'Maria
Pontes'))
AFTER lname;
update employee
SET  boss = 'Maria Pontes‘
                WHERE job_lvl > 150;
update employee
SET boss = 'Francisco Chang‘
                WHERE job_lvl <= 150;
```

VIEW RESULT(NOT ALL):



	emp_id	fname	minit	lname	boss	job_id	job_lvl	pub_id	hire_date
▶	A-C71970F	Aria		Cruz	Francisco Chang	10	87	1389	1991-10-26 00:00:00
	A-R89858F	Annette		Roulet	Maria Pontes	6	152	9999	1990-02-21 00:00:00
	AMD15433F	Ann	M	Devon	Maria Pontes	3	200	9952	1991-07-16 00:00:00
	ARD36773F	Anabela	R	Domingues	Francisco Chang	8	100	0877	1993-01-27 00:00:00
	CFH28514M	Carlos	F	Hernadez	Maria Pontes	5	211	9999	1989-04-21 00:00:00
	CGS88322F	Carine	G	Schmitt	Francisco Chang	13	64	1389	1992-07-07 00:00:00
	DBT39435M	Daniel	B	Tonini	Francisco Chang	11	75	0877	1990-01-01 00:00:00
	DWR65030M	Diego	W	Roel	Maria Pontes	6	192	1389	1991-12-16 00:00:00
	ENL44273F	Elizabeth	N	Lincoln	Francisco Chang	14	35	0877	1990-07-24 00:00:00
	F-C16315M	Francisco		Chang	Maria Pontes	4	227	9952	1990-11-03 00:00:00
	GHT50241M	Gary	H	Thomas	Maria Pontes	9	170	0736	1988-08-09 00:00:00
	H-B39728F	Helen		Bennett	Francisco Chang	12	35	0877	1989-09-21 00:00:00
	HAN90777M	Helvetius	A	Nagy	Francisco Chang	7	120	9999	1993-03-19 00:00:00
	HAS54740M	Howard	A	Snyder	Francisco Chang	12	100	0736	1988-11-19 00:00:00
	JYL26161F	Janine	Y	Labrune	Maria Pontes	5	172	9901	1991-05-26 00:00:00
	KFJ64308F	Karin	F	Josephs	Francisco Chang	14	100	0736	1992-10-17 00:00:00

CODE RESULT:



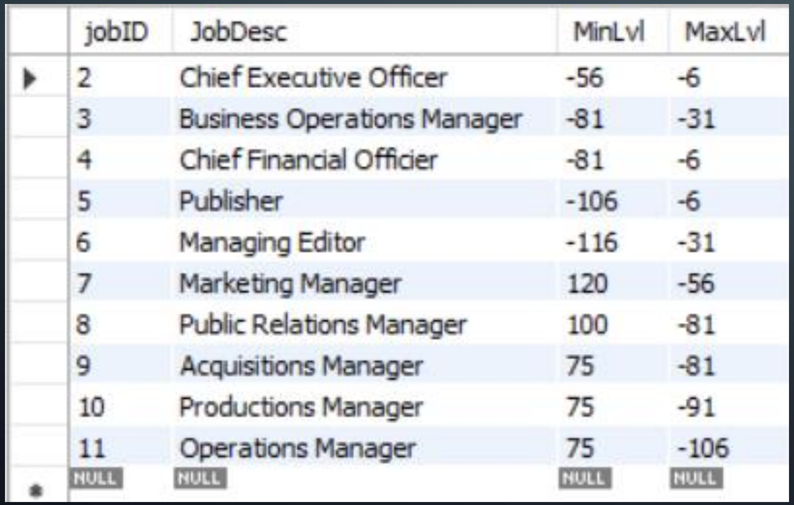
	emp_id	fname	minit	lname	boss	job_id	job_lvl	pub_id	hire_date
▶	A-C71970F	Aria		Cruz	Francisco Chang	10	87	1389	1991-10-26
	A-R89858F	Annette		Roulet	Maria Pontes	6	152	9999	1990-02-21
	AMD15433F	Ann	M	Devon	Maria Pontes	3	200	9952	1991-07-16
	ARD36773F	Anabela	R	Domingues	Francisco Chang	8	100	0877	1993-01-27
	CFH28514M	Carlos	F	Hernadez	Maria Pontes	5	211	9999	1989-04-21
	CGS88322F	Carine	G	Schmitt	Francisco Chang	13	64	1389	1992-07-07
	DBT39435M	Daniel	B	Tonini	Francisco Chang	11	75	0877	1990-01-01
	DWR65030M	Diego	W	Roel	Maria Pontes	6	192	1389	1991-12-16
	ENL44273F	Elizabeth	N	Lincoln	Francisco Chang	14	35	0877	1990-07-24
	F-C16315M	Francisco		Chang	Maria Pontes	4	227	9952	1990-11-03
	GHT50241M	Gary	H	Thomas	Maria Pontes	9	170	0736	1988-08-09
	H-B39728F	Helen		Bennett	Francisco Chang	12	35	0877	1989-09-21

Q2-7: MIGRATING THE JOBS DATA INTO JOBR WHICH HAS INTEGRITY CONSTRAINTS AND HOPE TO MAKE QUERY FASTER FROM JOBR

Optimizing (Using Integrity Constrains):

```
create table JobR (jobID int Primary Key, JobDesc varchar(50) unique,
MinLvl tinyint not null, MaxLvl tinyint not null);insert into JobR select * from
jobs where max_lvl > 100;select * from JobR;
```

CODE RESULT:



jobID	JobDesc	MinLvl	MaxLvl
2	Chief Executive Officer	-56	-6
3	Business Operations Manager	-81	-31
4	Chief Financial Officier	-81	-6
5	Publisher	-106	-6
6	Managing Editor	-116	-31
7	Marketing Manager	120	-56
8	Public Relations Manager	100	-81
9	Acquisitions Manager	75	-81
10	Productions Manager	75	-91
11	Operations Manager	75	-106
NULL	NULL	NULL	NULL

Conclusion:

11 different queries in term of meaning are designed, for the seek of optimization some of queries are duplicated having the same meaning but different syntax, however the execution time differences are not noticeable for the same queries since the queries run on local host with small number of records, but we hope that view creation, indexing, integrity constrain and, smart syntax would optimize the queries when it comes to a huge amount of data.