DATABASES Lab 1

Akshay Gurudath (Aksgu350), Keshav Padiyar Manuru (Kespa139)

April 19, 2021

Q1. List all employees, i.e. all tuples in the jbemployee relation.

```
select e1.name as Manager,
e2.name as Employee,
e2.salary,
e2.birthyear,
e2.startyear
from jbemployee e1,
jbemployee e2
where e2.manager = e1.id;
```

Manager	Employee	salary	birthyear	startyear
Thompson, Bob	Raveen, Lemont	11985	1950	1974
Thompson, Bob	Schmidt, Herman	11204	1936	1956
Thompson, Bob	Wallace, Maggi...	7880	1940	1959
Thomas, Tom	Brunet, Paul C.	17674	1938	1959
Thomas, Tom	Iwano, Masahiro	15641	1944	1970
Thomas, Tom	Zugnoni, Arthur...	19868	1928	1949
Smythe, Carol	Evans, Michael	5000	1952	1974
Smythe, Carol	Bailey, Chas M.	8377	1956	1975
Ross, Stanley	Collins, Joanne	7000	1950	1971
James, Mary	Choy, Wanda	11160	1947	1970
James, Mary	Bono, Sonny	13621	1939	1963
Hayes, Evelyn	Smith, Paul	6000	1952	1973
Hayes, Evelyn	Schwarz, Jason B.	13374	1944	1959
Edwards, Peter	Onstad, Richard	8779	1952	1971
Bullock, J.D.	Ross, Stanley	15908	1927	1945
Bullock, J.D.	Edwards, Peter	9000	1928	1958
Bullock, J.D.	Thompson, Bob	13000	1930	1970
Bullock, J.D.	Smythe, Carol	9050	1929	1967
Bullock, J.D.	Hayes, Evelyn	10100	1931	1963
Bullock, J.D.	James, Mary	12000	1920	1969
Bullock, J.D.	Williams, Judy	9000	1935	1969
Bullock, J.D.	Thomas, Tom	10000	1941	1962
Bullock, J.D.	Jones, Tim	12000	1940	1960

Q2. List the name of all departments in alphabetical order. Note: by "name" we mean the name attribute for all tuples in the jbdept relation.

select * from jbdept order by name;

id	name	store	floor	manager
1	Bargain	5	0	37
35	Book	5	1	55
10	Candy	5	1	13
43	Children's	8	2	32
73	Children's	5	1	10
19	Furniture	7	4	26
99	Giftwrap	5	1	98
14	Jewelry	8	1	33
47	Junior Miss	7	2	129
65	Junior's	7	3	37
26	Linens	7	3	157
20	Major App...	7	4	26
58	Men's	7	2	129
60	Sportswear	5	1	10
34	Stationary	5	1	33
49	Toys	8	2	35
28	Women's	8	2	32
63	Women's	7	3	32
70	Women's	5	1	10

Q3. What parts are not in store, i.e. qoh = 0? (qoh = Quantity On Hand)
select * from jbparts where qoh = 0;

select * from jbdept order by name;

id	name	color	weight	qoh
11	card reader	gray	327	0
12	card punch	gray	427	0
13	paper tape reader	black	107	0
14	paper tape punch	black	147	0

Q4. Which employees have a salary between 9000 (included) and 10000 (included)?

select * from jbemployee
where salary between 9000 and 10000
order by salary;

id	name	salary	manager	birthyear	startyear
13	Edwards, Peter	9000	199	1928	1958
98	Williams, Judy	9000	199	1935	1969
32	Smythe, Carol	9050	199	1929	1967
129	Thomas, Tom	10000	199	1941	1962

Q5. What was the age of each employee when they started working (startyear)?

select Name,
birthyear,
startyear,
startyear-birthyear as age
from jbemployee;

Name	birthyear	startyear	age
Ross, Stanley	1927	1945	18
Ross, Stuart	1931	1932	1
Edwards, Peter	1928	1958	30
Thompson, Bob	1930	1970	40
Smythe, Carol	1929	1967	38
Hayes, Evelyn	1931	1963	32
Evans, Michael	1952	1974	22
Raveen, Lemont	1950	1974	24
James, Mary	1920	1969	49
Williams, Judy	1935	1969	34
Thomas, Tom	1941	1962	21
Jones, Tim	1940	1960	20
Bullock, J.D.	1920	1920	0
Collins, Joanne	1950	1971	21
Brunet, Paul C.	1938	1959	21
Schmidt, Herman	1936	1956	20
Iwano, Masahiro	1944	1970	26
Smith, Paul	1952	1973	21
Onstad, Richard	1952	1971	19
Zugnoni, Arthur...	1928	1949	21
Choy, Wanda	1947	1970	23
Wallace, Maggi...	1940	1959	19
Bailey, Chas M.	1956	1975	19
Bono, Sonny	1939	1963	24
Schwarz, Jason B.	1944	1959	15

Q6. Which employees have a last name ending with “son”?
Query 1:
select * from jbemployee
where name like ’%son,%’;

Query 2:
select * from jbemployee
where substring(name, 1, position(”,” IN Name)-1)like ’%son’;

id	name	salary	manager	birthyear	startyear
26	Thompson, Bob	13000	199	1930	1970

Q7. Which items (note items, not parts) have been delivered by a supplier called Fisher-Price? Formulate this query using a subquery in the where-clause.

```
select * from jbitem
where supplier in
(select id from jbsupplier
where lower(trim(name)) = "fisher-price");
```

id	name	dept	price	qoh	supplier
43	Maze	49	325	200	89
107	The 'Feel' Book	35	225	225	89
119	Squeeze Ball	49	250	400	89

Q8. Formulate the same query as above, but without a subquery.

```
select item.*,
supplier.name
from jbitem item inner join jbsupplier supplier
on item.supplier = supplier.id
where lower(trim(supplier.name)) = "fisher-price";
```

id	name	dept	price	qoh	supplier	name
43	Maze	49	325	200	89	Fisher-Price
107	The 'Feel' Book	35	225	225	89	Fisher-Price
119	Squeeze Ball	49	250	400	89	Fisher-Price

Q9. Show all cities that have suppliers located in them. Formulate this query using a subquery in the where-clause.

```
select * from jbcity
where id in
(select distinct city from jbsupplier);
```

id	name	state
10	Amherst	Mass
21	Boston	Mass
100	New York	NY
106	White Plains	Neb
118	Hickville	Okla
303	Atlanta	Ga
537	Madison	Wisc
609	Paxton	Ill
752	Dallas	Tex
802	Denver	Colo
841	Salt Lake City	Utah
900	Los Angeles	Calif
921	San Diego	Calif
941	San Francisco	Calif
981	Seattle	Wash

Q10. What is the name and color of the parts that are heavier than a card reader? Formulate this query using a subquery in the where-clause. (The SQL query must not contain the weight as a constant.)

```
select name,
color
from jbparts
where weight  > (
select weight from jbparts
where lower(trim(name)) = 'card reader'
) ;
```

name	color
disk drive	black
tape drive	black
line printer	yellow
card punch	gray

Q11. Formulate the same query as above, but without a subquery. (The query must not contain the weight as a constant.)

```
select p1.name,
p1.color
from jbparts p1,
jbparts p2
where p1.weight ¿ p2.weight and
lower(trim(p2.name)) = 'card reader' ;
```

name	color
card punch	gray
line printer	yellow
tape drive	black
disk drive	black

Q12. What is the average weight of black parts?

```
select color,
avg(weight) as avg_weight
from jbparts
where color = 'black';
```

.

color	avg_weight
black	347.2500

Q13. What is the total weight of all parts that each supplier in Massachusetts ("Mass") has delivered? Retrieve the name and the total weight for each of these suppliers. Do not forget to take the quantity of delivered parts into account. Note that one row should be returned for each supplier.

```
select supplier.name,
sum(parts.weight * supply.quan) total_weight
```

```
from jbsupplier supplier inner join jbcity city
on supplier.city = city.id
inner join jbsupply supply
on supply.supplier = supplier.id
inner join jbparts parts
on supply.part = parts.id
where lower(trim(city.state)) = "mass"
group by supplier.name;
```

name	total_weight
Fisher-Price	1135000
DEC	3120

Q14. Create a new relation (a table), with the same attributes as the table items using the CREATE TABLE syntax where you define every attribute explicitly (i.e. not as a copy of another table). Then fill the table with all items that cost less than the average price for items. Remember to define primary and foreign keys in your table!

```
DROP TABLE IF EXISTS jbitems CASCADE;

CREATE TABLE jbitems (
id INT,
name VARCHAR(20),
dept INT NOT NULL,
price INT,
qoh INT UNSIGNED /* or, if check constraints were enforced: INT CHECK
(qoh  >= 0)*/,
supplier INT NOT NULL,
CONSTRAINT pk_item PRIMARY KEY(id)) ENGINE=InnoDB;

ALTER TABLE jbitems ADD CONSTRAINT fk_item_dept_n FOREIGN KEY
(dept) REFERENCES jbdept(id);

ALTER TABLE jbitems ADD CONSTRAINT fk_item_supplier_n FOREIGN
KEY (supplier) REFERENCES jbsupplier(id);
```

```
INSERT INTO jbitems
(select * from jbitem
where price  < (select avg(price) from jbitem));

select * from jbitems;
```

id	name	dept	price	qoh	supplier
11	Wash Cloth	1	75	575	213
19	Bellbottoms	43	450	600	33
21	ABC Blocks	1	198	405	125
23	1 lb Box	10	215	100	42
25	2 lb Box, Mix	10	450	75	42
26	Earrings	14	1000	20	199
43	Maze	49	325	200	89
106	Clock Book	49	198	150	125
107	The 'Feel' Book	35	225	225	89
118	Towels, Bath	26	250	1000	213
119	Squeeze Ball	49	250	400	89
120	Twin Sheet	26	800	750	213
165	Jean	65	825	500	33
258	Shirt	58	650	1200	33