Lab 1, Hong Zhang (honzh073) and BingRui Yao (binya518)

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

(Tuples are the records in the table, which are also the rows in the table.)

Use the following command:

select * from jbemployee;

Here is the command result:

#	id	name	salary	manager	birthyear	startyear
1	10	Ross, Stanley	15908	199	1927	1945
2	11	Ross, Stuart	12067	NULL	1931	1932
3	13	Edwards, Peter	9000	199	1928	1958
4	26	Thompson, Bob	13000	199	1930	1970
5	32	Smythe, Carol	9050	199	1929	1967
6	33	Hayes, Evelyn	10100	199	1931	1963
7	35	Evans, Michael	5000	32	1952	1974
8	37	Raveen, Lemont	11985	26	1950	1974
9	55	James, Mary	12000	199	1920	1969
10	98	Williams, Judy	9000	199	1935	1969
11	129	Thomas, Tom	10000	199	1941	1962
12	157	Jones, Tim	12000	199	1940	1960
13	199	Bullock, J.D.	27000	NULL	1920	1920
14	215	Collins, Joanne	7000	10	1950	1971
15	430	Brunet, Paul C.	17674	129	1938	1959
16	843	Schmidt, Her…	11204	26	1936	1956
17	994	Iwano, Masahiro	15641	129	1944	1970
18	1110	Smith, Paul	6000	33	1952	1973
19	1330	Onstad, Richard	8779	13	1952	1971
20	1523	Zugnoni, Arth…	19868	129	1928	1949
21	1639	Choy, Wanda	11160	55	1947	1970
22	2398	Wallace, Mag…	7880	26	1940	1959
23	4901	Bailey, Chas M.	8377	32	1956	1975
24	5119	Bono, Sonny	13621	55	1939	1963
25	5219	Schwarz, Jas…	13374	33	1944	1959
*	NULL	NULL	NULL	NULL	NULL	NULL

jbemployee 56 ×

2) 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	1	Bargain	5	0	37
2	35	Book	5	1	55
3	10	Candy	5	1	13
4	73	Children's	5	1	10
5	43	Children's	8	2	32
6	19	Furniture	7	4	26
7	99	Giftwrap	5	1	98
8	14	Jewelry	8	1	33
9	47	Junior Miss	7	2	129
10	65	Junior's	7	3	37
11	26	Linens	7	3	157
12	20	Major Ap…	7	4	26
13	58	Men's	7	2	129
14	60	Sportswear	5	1	10
15	34	Stationary	5	1	33
16	49	Toys	8	2	35
17	63	Women's	7	3	32
18	70	Women's	5	1	10
19	28	Women's	8	2	32
*	NULL	NULL	NULL	NULL	NULL

jbdept 57 ×

3) What parts are not in store, i.e. qoh = 0? (qoh = Quantity On Hand)

select * from jbparts where qoh = 0;

#	id	name	color	weight	qoh
1	11	card reader	gray	327	0
2	12	card punch	gray	427	0
3	13	paper tape reader	black	107	0
4	14	paper tape punch	black	147	0
*	NULL	NULL	NULL	NULL	NULL

jbparts 62 ×

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

select * from jbemployee where salary >= 9000 and salary <= 10000;

(another command:

select * from jbemployee where salary between 9000 and 10000;

)

#	id	name	color	weight	qoh
1	11	card reader	gray	327	0
2	12	card punch	gray	427	0
3	13	paper tape reader	black	107	0
4	14	paper tape punch	black	147	0
*	NULL	NULL	NULL	NULL	NULL

jbparts 58 ×

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

(age = startyear – birthyear)

select name, startyear-birthyear as age_started from jbemployee;

#	name	age_started
1	Ross, Stanley	18
2	Ross, Stuart	1
3	Edwards, Peter	30
4	Thompson, Bob	40
5	Smythe, Carol	38
6	Hayes, Evelyn	32
7	Evans, Michael	22
8	Raveen, Lemont	24
9	James, Mary	49
10	Williams, Judy	34
11	Thomas, Tom	21
12	Jones, Tim	20
13	Bullock, J.D.	0
14	Collins, Joanne	21
15	Brunet, Paul C.	21
16	Schmidt, Her…	20
17	Iwano, Masahiro	26
18	Smith, Paul	21
19	Onstad, Richard	19
20	Zugnoni, Arth…	21
21	Choy, Wanda	23
22	Wallace, Mag…	19
23	Bailey, Chas M.	19
24	Bono, Sonny	24
25	Schwarz, Jas…	15

Result 61 ×

6) Which employees have a last name ending with “son”?

select name from jbemployee where name like "%son,%";

#	name
1	Thompson, Bob

jbemployee 16 ×

7) 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 = (select id from jbsupplier where name = "Fisher-Price");

#	id	name	dept	price	qoh	supplier
1	43	Maze	49	325	200	89
2	107	The 'Feel' Book	35	225	225	89
3	119	Squeeze Ball	49	250	400	89
*	NULL	NULL	NULL	NULL	NULL	NULL

jbitem 17 ×

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

```
select jbitem.* from jbitem, jbsupplier
where jbsupplier.name = "Fisher-Price" and jbsupplier.id = jbitem.supplier;
```

#	id	name	dept	price	qoh	supplier
1	43	Maze	49	325	200	89
2	107	The 'Feel' Book	35	225	225	89
3	119	Squeeze Ball	49	250	400	89

Result 44 ×

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

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

#	id	name	state
1	10	Amherst	Mass
2	21	Boston	Mass
3	100	New York	NY
4	106	White Plains	Neb
5	118	Hickville	Okla
6	303	Atlanta	Ga
7	537	Madison	Wisc
8	609	Paxton	Ill
9	752	Dallas	Tex
10	802	Denver	Colo
11	841	Salt Lake ...	Utah
12	900	Los Angeles	Calif
13	921	San Diego	Calif
14	941	San Franc...	Calif
15	981	Seattle	Wash
*	NULL	NULL	NULL

jbcity 54 ×

10) 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.)

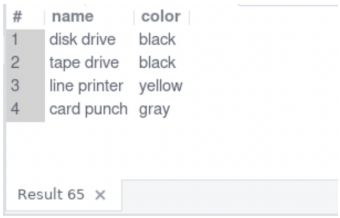
```
select name, color from jbparts
where weight > (select weight from jbparts where name = "card reader");
```

#	name	color
1	disk drive	black
2	tape drive	black
3	line printer	yellow
4	card punch	gray

jbparts 63 ×

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

```
select a.name, a.color from jbparts a, jbparts b
where a.weight > b.weight and b.name = "card reader";
```



#	name	color
1	disk drive	black
2	tape drive	black
3	line printer	yellow
4	card punch	gray

Result 65 ×

12) What is the average weight of black parts?

```
select avg(weight) from jbparts where color = "black";
```

#	avg(weight)
1	347.2500

Result 69 ×

13) 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 s.id, s.name, sum(p.weight * sy.quan) from jbsupplier s, jbparts p, jbsupply sy, jbcity c
where p.id = sy.part and sy.supplier = s.id and s.city = c.id and c.state = "Mass" group by s.id;
```

#	id	name	sum(p.weight * sy.quar
1	89	Fisher-Price	1135000
2	475	DEC	3120

Result 92 ×

14) 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 jbnewitem;

/*create an empty table ibnewitem*/
create table jbnewitem (id int(11) primary key, name varchar(20), dept int(11), price int(11),
qoh int(10), supplier int(10));

select * from jbnewitem;

/*insert items into table jbnewitem*/
insert into jbnewitem(id, name, dept, price, qoh, supplier)
select id, name, dept, price, qoh, supplier from jbitem
where jbitem.price < (select avg(price) from jbitem);

select * from jbnewitem;
```

#	id	name	dept	price	qoh	supplie
1	11	Wash Cloth	1	75	575	213
2	19	Bellbottoms	43	450	600	33
3	21	ABC Blocks	1	198	405	125
4	23	1 lb Box	10	215	100	42
5	25	2 lb Box, Mix	10	450	75	42
6	26	Earrings	14	1000	20	199
7	43	Maze	49	325	200	89
8	106	Clock Book	49	198	150	125
9	107	The 'Feel' Book	35	225	225	89
10	118	Towels, Bath	26	250	1000	213
11	119	Squeeze Ball	49	250	400	89
12	120	Twin Sheet	26	800	750	213
13	165	Jean	65	825	500	33
14	258	Shirt	58	650	1200	33
*	NULL	NULL	NULL	NULL	NULL	NULL

jbnewitem 100 ×