

## Big Data Analytics

Lab1, Aswath Mandakath Gopinath(aswma317), Singapura Ravi Varun(varsi146)

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

Ans.

Mysql > select \*

from jbemployee;

Output:

	id	name	salary	manager	birthyear	startyear
►	10	Ross, Stanley	15908	199	1927	1945
	11	Ross, Stuart	12067	NULL	1931	1932
	13	Edwards, Peter	9000	199	1928	1958
	26	Thompson, Bob	13000	199	1930	1970
	32	Smythe, Carol	9050	199	1929	1967
	33	Hayes, Evelyn	10100	199	1931	1963
	35	Evans, Michael	5000	32	1952	1974
	37	Raveen, Lemont	11985	26	1950	1974
	55	James, Mary	12000	199	1920	1969
	98	Williams, Judy	9000	199	1935	1969
	129	Thomas, Tom	10000	199	1941	1962
	157	Jones, Tim	12000	199	1940	1960
	199	Bullock, J.D.	27000	NULL	1920	1920
	215	Collins, Joanne	7000	10	1950	1971
	430	Brunet, Paul C.	17674	129	1938	1959
	843	Schmidt, Herman	11204	26	1936	1956
	994	Iwano, Masahiro	15641	129	1944	1970
	1110	Smith, Paul	6000	33	1952	1973
	1330	Onstad, Richard	8779	13	1952	1971
	1523	Zugnoni, Arthur A.	19868	129	1928	1949
	1639	Choy, Wanda	11160	55	1947	1970
	2398	Wallace, Maggie J.	7880	26	1940	1959
	4901	Bailey, Chas M.	8377	32	1956	1975
	5119	Bono, Sonny	13621	55	1939	1963
	5219	Schwarz, Jason B.	13374	33	1944	1959

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.

Ans.

Mysql>select distinct name

from jbdept

order by name asc;

Output:

	name
►	Bargain
	Book
	Candy
	Children's
	Furniture
	Giftwrap
	Jewelry
	Junior Miss
	Junior's
	Linens
	Major App...
	Men's
	Sportswear
	Stationary
	Toys
	Women's

Q3. What parts are not in store, i.e. qoh = 0? (qoh = Quantity On Hand)?

Ans.

```
Mysql> select *  
        from jbparts  
        where qoh = 0;
```

Output:

	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)?

Ans.

```
Mysql> select *  
        from jbemployee  
        where salary between 9000 and 10000;
```

Output:

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

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

Ans.

```
Mysql> select name, birthyear, startyear, (startyear-birthyear) as age
      from jbemployee;
```

Output:

	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 A.	1928	1949	21
	Choy, Wanda	1947	1970	23
	Wallace, Maggie J.	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”?

Ans.

```

Mysql>select name
      from jbemployee
      where substring_index(name, ' ', 1) like '%son%';

```

Output:

	name
▶	Thompson, Bob

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.

Ans.

```

Mysql>select *
      from jbitem
      where supplier in (select id
                        from jbsupplier
                        where name = 'Fisher-Price');

```

Output:

	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.

Ans.

```

Mysql>select *
      from jbitem, jbsupplier
      where jbsupplier.name = 'Fisher-Price' and jbitem.supplier = jbsupplier.id;

```

Output:

	id	name	dept	price	qoh	supplier	id	name	city
▶	43	Maze	49	325	200	89	89	Fisher-Price	21
	107	The 'Feel' Book	35	225	225	89	89	Fisher-Price	21
	119	Squeeze Ball	49	250	400	89	89	Fisher-Price	21

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

Ans.

```
Mysql>select *  
      from jbcity  
      where id in (select city  
                  from jbsupplier  
                  where jbcity.id = city);
```

Output:

	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.)

Ans.

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

Output:

	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.)

Ans.

Mysql>

```
select p2.name, p2.color
from jbparts p1, jbparts p2
where p1.name = 'card reader' and p2.weight > p1.weight;
```

Output:

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

Q12. What is the average weight of black parts?

Ans.

Mysql>

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

Output:

	color	avg_wt
▶	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.

Ans.

Mysql>

```
select id, name, sum(tot_wt) as sum_tot_wt from
(select suplr.id, suplr.name, sup.quan*parts.weight as tot_wt
from jbsupplier as suplr, jbcity as city,
jbsupply as sup, jbparts as parts
where suplr.city = city.id and city.state = 'Mass'
and sup.part = parts.id and suplr.id = sup.supplier) as tot_wt_tbl
group by name;
```

Output:

	id	name	sum_tot_wt
►	475	DEC	3120
	89	Fisher-Price	1135000

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!

```
create table jblowprice (
    id INT NOT NULL,
    name varchar(20),
    dept INT NOT NULL,
    price INT UNSIGNED,
    qoh INT UNSIGNED,
    supplier INT NOT NULL,
    primary key(id),
    foreign key (dept) references jbdept(id),
    foreign key (supplier) references jbsupplier(id));
```

Output:

	Field	Type	Null	Key	Default	Extra
►	id	int(11)	NO	PRI	NULL	
	name	varchar(20)	YES		NULL	
	dept	int(11)	NO	MUL	NULL	
	price	int(10) unsigned	YES		NULL	
	qoh	int(10) unsigned	YES		NULL	
	supplier	int(11)	NO	MUL	NULL	

insert into jblowprice

select \* from jbitem

where price < (select avg(price) from jbitem);

select \* from jblowprice;

Output:

	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