Lab compendium Lab 1

zahra jalilpour

2021-04-07

Contents

Lab 1 SQL-Queries and Views

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

```
SELECT *
  FROM jbemployee;
```

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, Herma	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	19868	129	1928	1949
1639	Choy, Wanda	11160	55	1947	1970
2398	Wallace, Maggie	7880	26	1940	1959
4901	Bailey, Chas M.	8377	32	1956	1975
5119	Bono, Sonny	13621	55	1939	1963
5219	Schwarz, Jason	13374	33	1944	1959

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 name
  FROM jbdept
  ORDER BY name;
```

	name
1	Bargain
2	Book
3	Candy
4	Children's
5	Children's
6	Furniture
7	Giftwrap
8	Jewelry
9	Junior Miss
10	Junior's
11	Linens
12	Major Appliances
13	Men's
14	Sportswear
15	Stationary
16	Toys
17	Women's
18	Women's
19	Women's
20	

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

```
SELECT name
  FROM jbparts
  WHERE jbparts.qoh=0;
```

	name
1	card reader
2	card punch
3	paper tape reader
4	paper tape punch
5	

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

```
SELECT name
  FROM  jbemployee
  WHERE  jbemployee.salary BETWEEN 9000 AND 10000;
```

	name
2	Edwards, Peter
3	Smythe, Carol
4	Williams, Judy
5	Thomas, Tom

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

```
SELECT name, (jbemployee.startyear - jbemployee.birthyear) AS age
  FROM jbemployee;
```

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

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

```
SELECT name
  FROM jbemployee
  WHERE name LIKE '%son,%';
```

	name
2	Thompson, Bob

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 AS itm
  WHERE itm.supplier=(SELECT sup.id
                      FROM jbsupplier AS sup
                      WHERE sup.name='Fisher-Price');
```

1	id	name	dept	price	qoh	supplier
2	43	Maze	49	325	200	89
3	107	The 'Feel' Book	35	225	225	89
4	119	Squeeze Ball	49	250	400	89

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

```
SELECT itm.id, itm.name, itm.dept, itm.price,itm.qoh,itm.supplier
  FROM jbitem AS itm, jbsupplier AS sup
  WHERE itm.supplier=sup.id AND sup.name='Fisher-Price';
```

1	id	name	dept	price	qoh	supplier
2	43	Maze	49	325	200	89
3	107	The 'Feel' Book	35	225	225	89
4	119	Squeeze Ball	49	250	400	89

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

```
SELECT *
  FROM jbcity AS cty
  WHERE cty.id IN (SELECT sup.city
  FROM jbsupplier AS sup);
```

	id	name	state
2	10	Amherst	Mass
3	21	Boston	Mass
4	100	New York	NY
5	106	White Plains	Neb
6	118	Hickville	Okla
7	303	Atlanta	Ga
8	537	Madison	Wisc
9	609	Paxton	Ill
10	752	Dallas	Tex
11	802	Denver	Colo
12	841	Salt Lake City	Utah
13	900	Los Angeles	Calif
14	921	San Diego	Calif
15	941	San Francisco	Calif
16	981	Seattle	Wash

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 prts.name , prts.color
  FROM jbparts AS prts
  WHERE prts.weight>(SELECT prts.weight
  FROM jbparts AS prts
  WHERE prts.name ='card reader');
```

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

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

```
SELECT prts.name, prts.color
  FROM jbparts AS prts, jbparts AS prts2
  WHERE prts2.name='card reader' AND prts.weight>prts2.weight;
```

	name	color
1	disk drive	black
2	tape drive	black
3	line printer	yellow
4	card punch	gray
5		

12)What is the average weight of black parts?

```
SELECT avg(prts.weight) AS avg_weight
  FROM jbparts AS prts WHERE prts.color='black';
```

	avg_weight
1	347.25
2	

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 sup.name, SUM(prts.weight*sp.quan) AS total_weight
  FROM jbparts AS prts, jbsupplier AS sup, jbsupply AS sp, jbcity AS cty
  WHERE prts.id=sp.part AND sup.id=sp.supplier
  AND sup.city= cty.id AND cty.state= 'Mass'
  GROUP BY sup.id;
```

	name	total_weight
1	Fisher-Price	1135000
2	DEC	3120
3		
4		

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!

```
CREATE TABLE jbitem_new
    (
        id INT,
        name VARCHAR(20),
```

```
        price INT,
        qoh INT UNSIGNED,
        dept INT,
        supplier INT,
        PRIMARY KEY (id),
        FOREIGN KEY (dept) REFERENCES jbdept(id),
        FOREIGN KEY (supplier) REFERENCES jbsupplier(id)
        );
 INSERT INTO jbitem_new(id, name, price, qoh, dept, supplier)
    SELECT itm.id, itm.name, itm.price, itm.qoh, itm.dept, itm.supplier
    FROM jbitem AS itm
    WHERE (itm.price)<(SELECT AVG(jbitem.price) FROM jbitem);

SELECT *
FROM  jbitem_new;
```

1	id	name	price	qoh	dept	supplier
2	11	Wash Cloth	75	575	1	213
3	19	Bellbottoms	450	600	43	33
4	21	ABC Blocks	198	405	1	125
5	23	1 lb Box	215	100	10	42
6	25	2 lb Box, Mix	450	75	10	42
7	26	Earrings	1000	20	14	199
8	43	Maze	325	200	49	89
9	106	Clock Book	198	150	49	125
10	107	The 'Feel' Book	225	225	35	89
11	118	Towels, Bath	250	1000	26	213
12	119	Squeeze Ball	250	400	49	89
13	120	Twin Sheet	800	750	26	213
14	165	Jean	825	500	65	33
15	258	Shirt	650	1200	58	33