Big Data Analystics - Lab 01

Julius Kittler (anner218) and Maximilian Pfundstein (maxpf364)

2019-03-27

Contents

1 SQL-Statements **1**
1.1 Statement 1 . . . 1
1.2 Statement 2 . . . 2
1.3 Statement 3 . . . 3
1.4 Statement 4 . . . 3
1.5 Statement 5 . . . 3
1.6 Statement 6 . . . 4
1.7 Statement 7 . . . 4
1.8 Statement 8 . . . 4
1.9 Statement 9 . . . 5
1.10 Statement 10 . . . 5
1.11 Statement 11 . . . 5
1.12 Statement 12 . . . 6
1.13 Statement 13 . . . 6
1.14 Statement 14 . . . 6

1 SQL-Statements

1.1 Statement 1

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	NA	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	NA	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

1.2 Statement 2

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
Bargain
Book
Candy
Children's
Children's
Furniture
Giftwrap
Jewelry
Junior Miss
Junior's
Linens
Major Appliances
Men's
Sportswear
Stationary
Toys
Women's
Women's
Women's

1.3 Statement 3

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

```
SELECT name FROM jbparts WHERE qoh = 0 ORDER BY name
```

name
card punch
card reader
paper tape punch
paper tape reader

1.4 Statement 4

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

```
SELECT name FROM jbemployee WHERE salary >= 9000 AND salary <= 10000
  ORDER BY name
```

name
Edwards, Peter
Smythe, Carol
Thomas, Tom
Williams, Judy

1.5 Statement 5

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

```
SELECT *, (startyear-birthyear) AS "Age When Started" FROM jbemployee
```

id	name	salary	manager	birthyear	startyear	Age.When.Started
10	Ross, Stanley	15908	199	1927	1945	18
11	Ross, Stuart	12067	NA	1931	1932	1
13	Edwards, Peter	9000	199	1928	1958	30
26	Thompson, Bob	13000	199	1930	1970	40
32	Smythe, Carol	9050	199	1929	1967	38
33	Hayes, Evelyn	10100	199	1931	1963	32
35	Evans, Michael	5000	32	1952	1974	22
37	Raveen, Lemont	11985	26	1950	1974	24
55	James, Mary	12000	199	1920	1969	49
98	Williams, Judy	9000	199	1935	1969	34
129	Thomas, Tom	10000	199	1941	1962	21
157	Jones, Tim	12000	199	1940	1960	20
199	Bullock, J.D.	27000	NA	1920	1920	0
215	Collins, Joanne	7000	10	1950	1971	21
430	Brunet, Paul C.	17674	129	1938	1959	21
843	Schmidt, Herman	11204	26	1936	1956	20
994	Iwano, Masahiro	15641	129	1944	1970	26
1110	Smith, Paul	6000	33	1952	1973	21
1330	Onstad, Richard	8779	13	1952	1971	19
1523	Zugnoni, Arthur A.	19868	129	1928	1949	21
1639	Choy, Wanda	11160	55	1947	1970	23
2398	Wallace, Maggie J.	7880	26	1940	1959	19
4901	Bailey, Chas M.	8377	32	1956	1975	19
5119	Bono, Sonny	13621	55	1939	1963	24
5219	Schwarz, Jason B.	13374	33	1944	1959	15

1.6 Statement 6

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

```
SELECT * FROM jbemployee WHERE SUBSTRING_INDEX(name, ',', 1) LIKE '%son';
```

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

1.7 Statement 7

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 IN
  (SELECT id from jbsupplier WHERE 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

1.8 Statement 8

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

```
SELECT * FROM jbitem INNER JOIN jbsupplier ON jbitem.supplier = jbsupplier.id
  WHERE jbsupplier.name = "Fisher-Price"
```

id	name	dept	price	qoh	supplier	id.1	name.1	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

1.9 Statement 9

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

```
SELECT * FROM jbcity WHERE jbcity.id IN (SELECT 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

1.10 Statement 10

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

1.11 Statement 11

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

```
SELECT parts1.name, parts1.color FROM jbparts AS parts1, jbparts AS parts2
  WHERE parts2.name = "card reader" AND parts1.weight > parts2.weight
```

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

1.12 Statement 12

12) What is the average weight of black parts?

```
SELECT AVG(weight) FROM jbparts WHERE color = 'black'
```

AVG.weight.
347.2500

1.13 Statement 13

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 jbsupplier.name, SUM(js.quan * jp.weight) AS total_weight FROM jbsupplier
  LEFT JOIN jbcity jc ON jbsupplier.city = jc.id
  LEFT JOIN jbsupply js ON jbsupplier.id = js.supplier
  LEFT JOIN jbparts jp on js.part = jp.id
  WHERE  jc.state = 'Mass'
  GROUP BY jbsupplier.id
```

name	total_weight
Fisher-Price	1135000
DEC	3120

1.14 Statement 14

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_custom (
    id INT,
    name VARCHAR(20),
    dept INT NOT NULL,
    price INT,
    qoh INT UNSIGNED,
    supplier INT,
    CONSTRAINT pk_item PRIMARY KEY(id),
    CONSTRAINT dept FOREIGN KEY(dept) REFERENCES jbdept(id),
    CONSTRAINT supplier FOREIGN KEY(supplier) REFERENCES jbsupplier(id));

INSERT INTO jbitem_custom (id, name, dept, price, qoh, supplier)
  SELECT ji.id, ji.name, ji.dept, ji.price, ji.qoh, ji.supplier
  FROM jbitem AS ji WHERE ji.price < (SELECT AVG(jbitem.price) FROM jbitem)
```