Lab 1. SQL-Queries and Views

Stefano Toffol (steto820) and Nahid Farazmand (nahfa911)

March 27, 2019

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

```sql
-- Q1. All employees
SELECT E.name, M.name AS manager, E.salary, E.birthyear, E.startyear
FROM jbemployee E , jbemployee M
WHERE E.manager = M.id
ORDER BY M.name
```

Table 1: 23 records

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

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.

```sql
-- Q2. List the name of all departments in alphabetical order
SELECT D.name, C.name AS store, D.floor, E.name AS manager
FROM jbdept AS D, jbstore AS S, jbemployee AS E, jbcity AS C
WHERE S.id = D.store
  AND D.manager = E.id
  AND S.city = C.id
ORDER BY D.name
```

Table 2: 19 records

name	store	floor	manager
Bargain	San Francisco	0	Raveen, Lemont
Book	San Francisco	1	James, Mary
Candy	San Francisco	1	Edwards, Peter
Children’s	San Francisco	1	Ross, Stanley
Children’s	El Cerrito	2	Smythe, Carol
Furniture	Oakland	4	Thompson, Bob
Giftwrap	San Francisco	1	Williams, Judy
Jewelry	El Cerrito	1	Hayes, Evelyn
Junior Miss	Oakland	2	Thomas, Tom
Junior’s	Oakland	3	Raveen, Lemont
Linens	Oakland	3	Jones, Tim
Major Appliances	Oakland	4	Thompson, Bob
Men’s	Oakland	2	Thomas, Tom
Sportswear	San Francisco	1	Ross, Stanley
Stationary	San Francisco	1	Hayes, Evelyn
Toys	El Cerrito	2	Evans, Michael
Women’s	San Francisco	1	Ross, Stanley
Women’s	El Cerrito	2	Smythe, Carol
Women’s	Oakland	3	Smythe, Carol

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

```sql
-- Q3. parts that are not in store
SELECT *
FROM jbparts
WHERE qoh = 0
```

Table 3: 4 records

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

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

```sql
-- Q4. employees with a salary between 9000 (included) and 10000 (included)
SELECT *
FROM jbemployee
WHERE salary BETWEEN 9000 and 10000
```

Table 4: 4 records

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

id	name	salary	manager	birthyear	startyear
129	Thomas, Tom	10000	199	1941	1962

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

```sql
-- Q5. The age of employees when they started working
SELECT name, startyear - birthyear AS Age
FROM jbemployee
```

Table 5: 25 records

name	Age
Ross, Stanley	18
Ross, Stuart	1
Edwards, Peter	30
Thompson, Bob	40
Smythe, Carol	38
Hayes, Evelyn	32
Evans, Michael	22
Raveen, Lemont	24
James, Mary	49
Williams, Judy	34
Thomas, Tom	21
Jones, Tim	20
Bullock, J.D.	0
Collins, Joanne	21
Brunet, Paul C.	21
Schmidt, Herman	20
Iwano, Masahiro	26
Smith, Paul	21
Onstad, Richard	19
Zugnoni, Arthur A.	21
Choy, Wanda	23
Wallace, Maggie J.	19
Bailey, Chas M.	19
Bono, Sonny	24
Schwarz, Jason B.	15

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

```sql
-- Q6. Employees that have a last name ending with "son"
SELECT *
FROM jbemployee
WHERE name LIKE '%son,%'
```

Table 6: 1 records

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

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.

```sql
-- Q7. items have been delivered by Fisher-Price (With subquery)
SELECT I.name, D.name, I.price, I.qoh
FROM jbitem I, jbdept D
WHERE supplier IN
               (SELECT id
                FROM jbsupplier
                WHERE name = 'Fisher-Price'
                     )
        AND I.dept = D.id
```

Table 7: 3 records

name	name	price	qoh
Maze	Toys	325	200
The 'Feel' Book	Book	225	225
Squeeze Ball	Toys	250	400

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

```sql
-- Q8. items have been delivered by Fisher-Price (Without subquery)
SELECT I.name, D.name, I.price, I.qoh
FROM jbitem I, jbdept D, jbsupplier S
WHERE I.supplier = S.id
        AND I.dept = D.id
        AND S.name = 'Fisher-Price'
```

Table 8: 3 records

name	name	price	qoh
Maze	Toys	325	200
The 'Feel' Book	Book	225	225
Squeeze Ball	Toys	250	400

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

```sql
-- Q9. All cities that have suppliers located in them
SELECT *
FROM jbcity
WHERE id IN
           (SELECT city
            FROM jbsupplier)
```

Table 9: 15 records

id	name	state
10	Amherst	Mass
21	Boston	Mass

id	name	state
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

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.)

```sql
-- Q10. The name and color of the parts that are heavier than a card reader
-- (With SubQuery)
SELECT name, color
FROM jbparts
WHERE weight >
              (SELECT weight
               FROM jbparts
               WHERE name = 'card reader')
```

Table 10: 4 records

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

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

```sql
-- Q11. The name and color of the parts that are heavier than a card reader
-- (Without SubQuery)
SELECT P.name, P.color
FROM jbparts AS P, jbparts AS CR
WHERE P.weight > CR.weight
      AND CR.name = 'card reader'
```

Table 11: 4 records

name	color
disk drive	black
tape drive	black

name	color
line printer	yellow
card punch	gray

12.What is the average weight of black parts?

```
-- Q12. The average weight of black parts
SELECT AVG(weight)
FROM jbparts
WHERE color = 'black'
GROUP BY color
```

Table 12: 1 records

AVG(weight)
347.25

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.

```
-- Q13. the total weight of all parts that each supplier
-- in Massachusetts ("Mass") has delivered
SELECT SP.name AS supplier, SUM(P.weight*qoh)
FROM jbsupply AS S, jbsupplier AS SP, jbparts AS P, jbcity AS C
WHERE S.supplier = SP.id
      AND S.part = P.id
      AND SP.city = C.id
      AND C.state = 'Mass'
GROUP BY SP.name
```

Table 13: 2 records

supplier	SUM(P.weight*qoh)
DEC	4470
Fisher-Price	3170

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!

```
-- Q14. A new relation (a table), with the same attributes
-- as the table items

-- Creating Table with PK
```

```
CREATE TABLE IF NOT EXISTS jbitems (
    id INT,
    name VARCHAR(20),
    dept INT NOT NULL,
    price INT,
    qoh INT UNSIGNED,
    supplier INT NOT NULL,
    CONSTRAINT pk_item PRIMARY KEY(id)) ENGINE=InnoDB;

-- Adding FKs
ALTER TABLE jbitems
ADD CONSTRAINT fk_itms_dpt
FOREIGN KEY (dept) REFERENCES jbdept(id);

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

-- Inserting data
INSERT INTO jbitems
(SELECT *
 FROM jbitem
 WHERE price < (SELECT AVG(price)
                FROM jbitem)
)
```

The table that we create

```
SELECT *
FROM jbitems
```

Table 14: 14 records

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