Lab 1

Mohammed Ali{mohal954} and Adesijibomi Aderinto{adead268}

4/2/2022

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, 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

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

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

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

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

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

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

1	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, startyear - birthyear AS start_age
FROM
    jbemployee;
```

1	name	start_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, Herman	20
18	Iwano, Masahiro	26
19	Smith, Paul	21
20	Onstad, Richard	19
21	Zugnoni, Arthur A.	21
22	Choy, Wanda	23
23	Wallace, Maggie J.	19
24	Bailey, Chas M.	19
25	Bono, Sonny	24
26	Schwarz, Jason B.	15

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

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

	name
1	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
    name
FROM
    jbitem
WHERE
    supplier IN (SELECT
            id
        FROM
            jbsupplier
        WHERE
            name = 'Fisher-Price');
```

	name
1	Maze
2	The 'Feel' Book
3	Squeeze Ball

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

```
SELECT
    jbitem.name AS item_name, jbsupplier.name AS supplier_name
FROM
    jbitem
        JOIN
    jbsupplier ON jbitem.supplier = jbsupplier.id
WHERE
    jbsupplier.name = 'Fisher-Price';
```

	item_name	supplier_name
1	Maze	Fisher-Price
2	The 'Feel' Book	Fisher-Price
3	Squeeze Ball	Fisher-Price

9) Show all cities that have suppliers located in them. Formulate this query using a

subquery in the where-clause.

```
SELECT
    name
FROM
    jbcity
WHERE
    id IN (SELECT
            city
        FROM
            jbsupplier);
```

	name
1	Amherst
2	Boston
3	New York
4	White Plains
5	Hickville
6	Atlanta
7	Madison
8	Paxton
9	Dallas
10	Denver
11	Salt Lake City
12	Los Angeles
13	San Diego
14	San Francisco
15	Seattle

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

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

contain the weight as a constant.)

```
SELECT
    cond1.name, cond1.color
FROM
    jbparts AS cond1,
    jbparts AS cond2
WHERE
    cond2.name = 'card reader'
        AND cond1.weight > cond2.weight;
```

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

12) What is the average weight of black parts?

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

	AvgWeights_black_Parts
1	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.

```
SELECT
    jbsupplier.name AS Supplier_Name,
    SUM(weight * quan) AS SumTotal_weight
FROM
    jbsupplier
        JOIN
    jbcity ON jbsupplier.city = jbcity.id
        JOIN
    jbsupply ON jbsupplier.id = jbsupply.supplier
        JOIN
    jbparts ON jbparts.id = jbsupply.part
WHERE
    state = 'Mass'
GROUP BY jbsupplier.name;
```

1	Supplier_Name	SumTotal_weight
2	DEC	3120
3	Fisher-Price	1135000

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 jbnewTableItem AS SELECT * FROM
    jbitem
WHERE
    price < (SELECT
            AVG(price)
        FROM
            jbitem);
ALTER TABLE jbnewTableItem ADD PRIMARY KEY (id);
ALTER TABLE jbnewTableItem ADD FOREIGN KEY (supplier) REFERENCES jbitem(supplier);
SELECT
    *
FROM
    jbnewtableitem
```

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