732A54 Big Data Analytics

Lab 1; SQL-Queries and Views

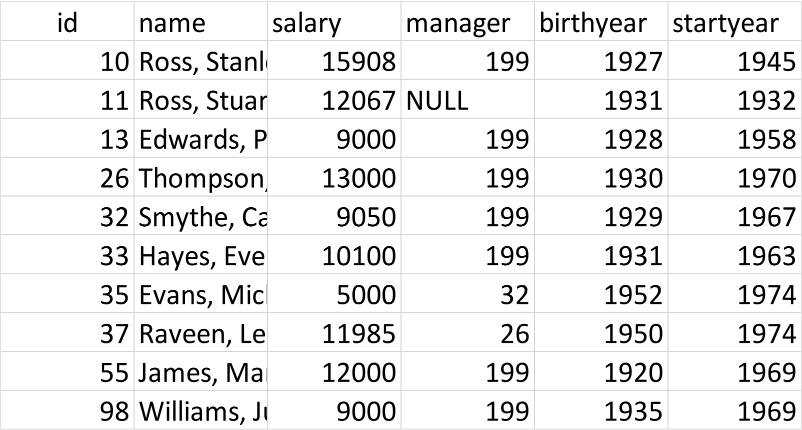
Group G6:

Hoda Fakharzadehjahromy(hodfa840)
Seyda Aqsa Iftekhar(syeif776)

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

```
SELECT * FROM jbemployee;
```

Output:

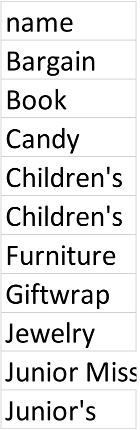


id	name	salary	manager	birthyear	startyear
10	Ross, Stanl	15908	199	1927	1945
11	Ross, Stuar	12067	NULL	1931	1932
13	Edwards, P	9000	199	1928	1958
26	Thompson,	13000	199	1930	1970
32	Smythe, Ca	9050	199	1929	1967
33	Hayes, Eve	10100	199	1931	1963
35	Evans, Mic	5000	32	1952	1974
37	Raveen, Le	11985	26	1950	1974
55	James, Ma	12000	199	1920	1969
98	Williams, Ju	9000	199	1935	1969

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;
```

Output:

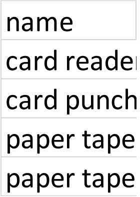


name
Bargain
Book
Candy
Children's
Children's
Furniture
Giftwrap
Jewelry
Junior Miss
Junior's

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

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

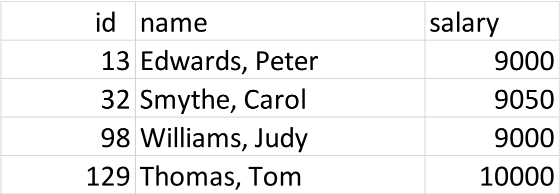
Output:



name
card reade
card punch
paper tape
paper tape

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

Output:

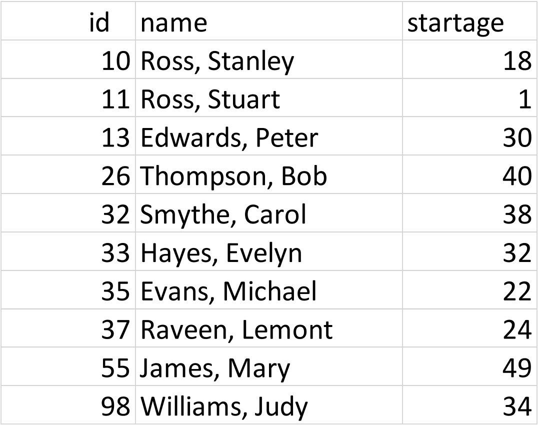


id	name	salary
13	Edwards, Peter	9000
32	Smythe, Carol	9050
98	Williams, Judy	9000
129	Thomas, Tom	10000

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

```sql
SELECT id,name, startyear - birthyear AS startage
 FROM jbemployee ;
```

Output:

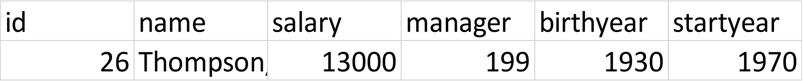


id	name	startage
10	Ross, Stanley	18
11	Ross, Stuart	1
13	Edwards, Peter	30
26	Thompson, Bob	40
32	Smythe, Carol	38
33	Hayes, Evelyn	32
35	Evans, Michael	22
37	Raveen, Lemont	24
55	James, Mary	49
98	Williams, Judy	34

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

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

Output:

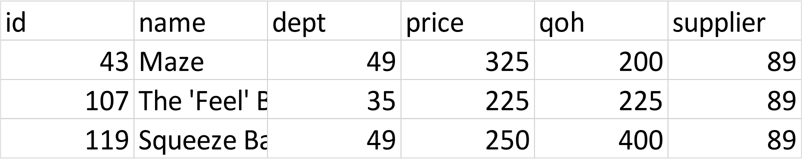


id	name	salary	manager	birthyear	startyear
26	Thompson,	13000	199	1930	1970

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.

```
SELECT * FROM jbitem WHERE supplier=(select id from jbsupplier where name="Fisher-Price");
```

Output:



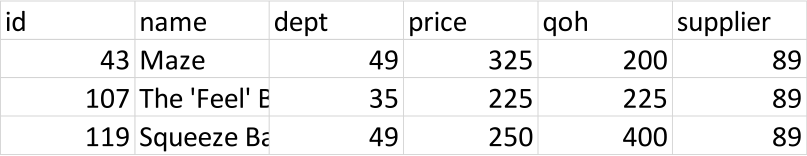
id	name	dept	price	qoh	supplier
43	Maze	49	325	200	89
107	The 'Feel' E	35	225	225	89
119	Squeeze Ba	49	250	400	89

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

```
select jbitem.* from jbsupplier
inner join jbitem
on jbitem.supplier = jbsupplier.id

where jbsupplier.name = 'Fisher-Price';
```

Output:



id	name	dept	price	qoh	supplier
43	Maze	49	325	200	89
107	The 'Feel' E	35	225	225	89
119	Squeeze Ba	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 WHERE id in (select city from jbsupplier);
```

Output:

id	name	state
10	Amherst	Mass
21	Boston	Mass
100	New York	NY
106	White Plair	Neb
118	Hickville	Okla
303	Atlanta	Ga
537	Madison	Wisc
609	Paxton	Ill
752	Dallas	Tex
802	Denver	Colo

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');
```

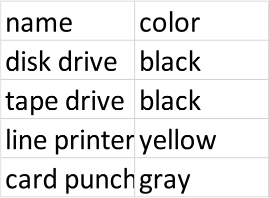
Output:

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.)

```
SELECT t1.name, t1.color
FROM jbparts t1,jbparts t2 WHERE t1.weight > t2.weight AND t2.name='card reader';
```

Output:

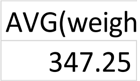


name	color
disk drive	black
tape drive	black
line printer	yellow
card punch	gray

12)What is the average weight of black parts?

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

Output:



AVG(weigh
347.25

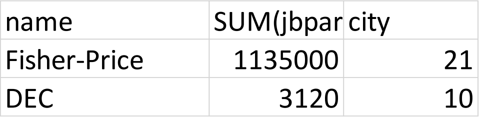
13)What is the total weight of all parts that each supplier in Massachusetts (“Mass”) has delivered? Retrieve the name and the total weight for each of these suppliers. Do not forget to take the quantity of delivered parts into account. Note that one row should be returned for each supplier.

```
SELECT jbsupplier.name,SUM(jbparts.weight*jbsupply.quan),jbsupplier.city
FROM  jbsupplier,jbsupply, jbparts,jbcity

WHERE jbcity.state = "Mass" AND
jbsupplier.city =jbcity.id AND
jbsupply.supplier = jbsupplier.id AND
jbparts.id = jbsupply.part

GROUP BY jbsupplier.id;
```

Output:



name	SUM(jbpar	city
Fisher-Price	1135000	21
DEC	3120	10

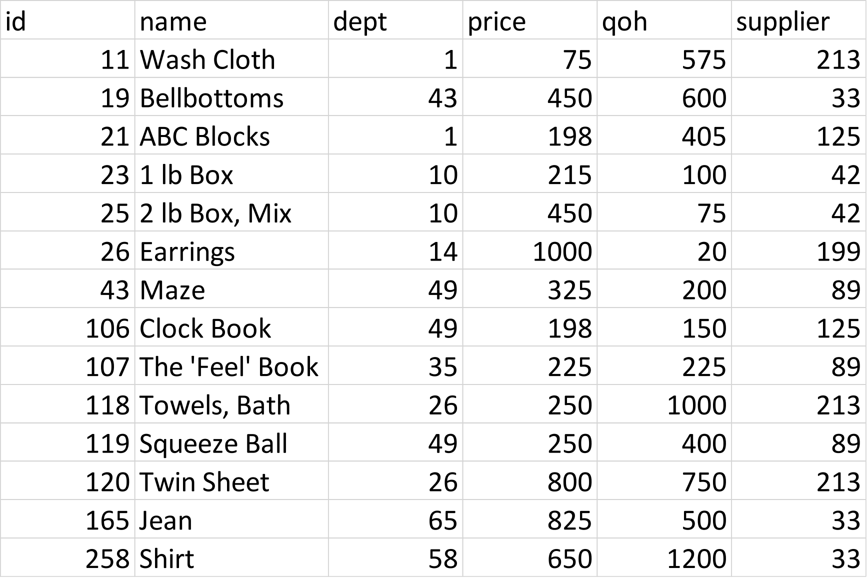
14) Create a new relation (a table), with the same attributes as the table items using the CREATE TABLE syntax where you define every attribute explicitly (i.e. not as a copy of another table). Then fill the table with all items that cost less than the average price for items. Remember to define primary and foreign keys in your table!

```
CREATE TABLE jblowpriceitem (
    id INT NOT NULL,
    name VARCHAR(20),
    dept INT NOT NULL,
    price INT NOT NULL,
    qoh INT UNSIGNED,
    supplier INT NOT NULL,
    CONSTRAINT pk_item PRIMARY KEY(id));


ALTER TABLE jblowpriceitem ADD CONSTRAINT fk_jblowpriceitem_dept FOREIGN KEY (dept) REFERENCES jbdept(id);
ALTER TABLE jblowpriceitem ADD CONSTRAINT fk_jblowpriceitem_supplier FOREIGN KEY (supplier) REFERENCES jbsupplier(id);


INSERT INTO jblowpriceitem (SELECT * FROM jbitem WHERE price< (SELECT AVG(price) FROM jbitem));
```

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