# Lab compendium Lab 1

*Phillip Hölscher (phiho267)*

*28 3 2019*

## 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 |
|---|---|---|---|---|---|---|
| 1 | 10 | Ross, Stanley | 15908 | 199 | 1927 | 1945 |
| 2 | 11 | Ross, Stuart | 12067 | <null> | 1931 | 1932 |
| 3 | 13 | Edwards, Peter | 9000 | 199 | 1928 | 1958 |
| 4 | 26 | Thompson, Bob | 13000 | 199 | 1930 | 1970 |
| 5 | 32 | Smythe, Carol | 9050 | 199 | 1929 | 1967 |
| 6 | 33 | Hayes, Evelyn | 10100 | 199 | 1931 | 1963 |
| 7 | 35 | Evans, Michael | 5000 | 32 | 1952 | 1974 |
| 8 | 37 | Raveen, Lemont | 11985 | 26 | 1950 | 1974 |
| 9 | 55 | James, Mary | 12000 | 199 | 1920 | 1969 |
| 10 | 98 | Williams, Judy | 9000 | 199 | 1935 | 1969 |
| 11 | 129 | Thomas, Tom | 10000 | 199 | 1941 | 1962 |
| 12 | 157 | Jones, Tim | 12000 | 199 | 1940 | 1960 |
| 13 | 199 | Bullock, J.D. | 27000 | <null> | 1920 | 1920 |
| 14 | 215 | Collins, Joanne | 7000 | 10 | 1950 | 1971 |
| 15 | 430 | Brunet, Paul C. | 17674 | 129 | 1938 | 1959 |
| 16 | 843 | Schmidt, Herman | 11204 | 26 | 1936 | 1956 |
| 17 | 994 | Iwano, Masahiro | 15641 | 129 | 1944 | 1970 |
| 18 | 1110 | Smith, Paul | 6000 | 33 | 1952 | 1973 |
| 19 | 1330 | Onstad, Richard | 8779 | 13 | 1952 | 1971 |
| 20 | 1523 | Zugnoni, Arthur A. | 19868 | 129 | 1928 | 1949 |
| 21 | 1639 | Choy, Wanda | 11160 | 55 | 1947 | 1970 |
| 22 | 2398 | Wallace, Maggie J. | 7880 | 26 | 1940 | 1959 |
| 23 | 4901 | Bailey, Chas M. | 8377 | 32 | 1956 | 1975 |
| 24 | 5119 | Bono, Sonny | 13621 | 55 | 1939 | 1963 |
| 25 | 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;
```

| | name |
|---|---|
| 1 | Edwards, Peter |
| 2 | Smythe, Carol |
| 3 | Williams, Judy |
| 4 | Thomas, Tom |

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

```
select name,  startyear-birthyear as age_started
from jbemployee;
```

| | name | age_started |
|---|---|---|
| 1 | Ross, Stanley | 18 |
| 2 | Ross, Stuart | 1 |
| 3 | Edwards, Peter | 30 |
| 4 | Thompson, Bob | 40 |
| 5 | Smythe, Carol | 38 |
| 6 | Hayes, Evelyn | 32 |
| 7 | Evans, Michael | 22 |
| 8 | Raveen, Lemont | 24 |
| 9 | James, Mary | 49 |
| 10 | Williams, Judy | 34 |
| 11 | Thomas, Tom | 21 |
| 12 | Jones, Tim | 20 |
| 13 | Bullock, J.D. | 0 |
| 14 | Collins, Joanne | 21 |
| 15 | Brunet, Paul C. | 21 |
| 16 | Schmidt, Herman | 20 |
| 17 | Iwano, Masahiro | 26 |
| 18 | Smith, Paul | 21 |
| 19 | Onstad, Richard | 19 |
| 20 | Zugnoni, Arthur A. | 21 |
| 21 | Choy, Wanda | 23 |
| 22 | Wallace, Maggie J. | 19 |
| 23 | Bailey, Chas M. | 19 |
| 24 | Bono, Sonny | 24 |
| 25 | 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, jbsupplier.name as supplier
from jbitem
join jbsupplier
on jbitem.supplier = jbsupplier.id
where jbsupplier.name = "Fisher-Price";
```

| | item | supplier |
|---|---|---|
| 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 jbstore);
```

| | name |
|---|---|
| 1 | San Francisco |
| 2 | El Cerrito |
| 3 | Oakland |

### 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 *
from jbparts
order by weight DESC
limit 4;
```

| | id | name | color | weight | qoh |
|---|---|---|---|---|---|
| 1 | 3 | disk drive | black | 685 | 2 |
| 2 | 6 | line printer | yellow | 578 | 3 |
| 3 | 4 | tape drive | black | 450 | 4 |
| 4 | 12 | card punch | gray | 427 | 0 |

### 12) What is the average weight of black parts?

```
select avg(weight) as "avg of weight"
from jbparts
where color = "black";
```

| | `avg of weight` |
|---|---|
| 1 | 347.2500 |

## 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,  sum(quan * weight) as weight_totalsum
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;
```

| | supplier | weight_totalsum |
|---|---|---|
| 1 | DEC | 3120 |
| 2 | Fisher-Price | 1135000 |

## 14) Create a new relation (a table), with the same attributes as the table items using the CREATE TABLE

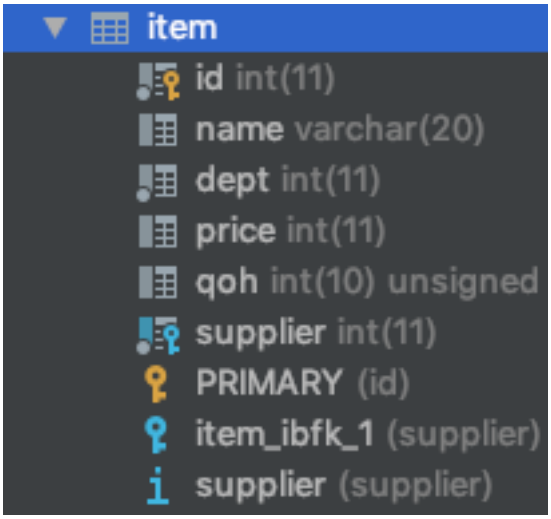
syntax where you define every attribute explicitly (i.e. not as a copy of another table). Then fill the table with all items that cost less than the average price for items. Remember to define primary and foreign keys in your table!

```
create table item as
select *
from jbitem
where price < (select avg(price)
                  from jbitem);

ALTER TABLE item
ADD PRIMARY KEY (id);

ALTER TABLE item
ADD FOREIGN KEY (supplier) REFERENCES jbitem(supplier);
```

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


item
id int(11)
name varchar(20)
dept int(11)
price int(11)
qoh int(10) unsigned
supplier int(11)
PRIMARY (id)
item_ibfk_1 (supplier)
supplier (supplier)