

lab1

```
library(RMySQL)
```

```
## Loading required package: DBI
```

```
mydb = dbConnect(MySQL(), user='andst745', password='andst745c880', dbname='andst745', host='mariadb.edu')
```

LAB1

Part 1

1.1

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

```
query = "select * from jbemployee;"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

##	id	name	salary	manager	birthyear	startyear
## 1	10	Ross, Stanley	15908	199	1927	1945
## 2	11	Ross, Stuart	12067	NA	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	NA	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

1.2

List the name of all departments in alphabetical order. Note: by name we mean the name attribute for all tuples in the *jbdept* relation.

```
query = "select name from jbdept order by name;"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##           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
```

1.3

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

```
query = "select * from jbparts where qoh=0;"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##  id           name color weight qoh
## 1 11      card reader  gray   327  0
## 2 12      card punch  gray   427  0
## 3 13 paper tape reader black   107  0
## 4 14 paper tape punch black   147  0
```

1.4

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

```
query = "select * from jbemployee where salary>=9000 and salary<=10000;"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##      id          name salary manager birthyear startyear
## 1  13 Edwards, Peter  9000      199      1928      1958
## 2  32 Smythe, Carol  9050      199      1929      1967
## 3  98 Williams, Judy  9000      199      1935      1969
## 4 129   Thomas, Tom 10000      199      1941      1962
```

1.5

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

```
query = "select id, name, startyear-birthyear start_age from jbemployee;"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##      id          name start_age
## 1   10      Ross, Stanley      18
## 2   11      Ross, Stuart       1
## 3   13  Edwards, Peter      30
## 4   26  Thompson, Bob       40
## 5   32   Smythe, Carol      38
## 6   33   Hayes, Evelyn      32
## 7   35  Evans, Michael      22
## 8   37  Raveen, Lemont      24
## 9   55      James, Mary      49
## 10  98  Williams, Judy      34
## 11 129   Thomas, Tom       21
## 12 157   Jones, Tim        20
## 13 199   Bullock, J.D.       0
## 14 215  Collins, Joanne      21
## 15 430  Brunet, Paul C.      21
## 16 843  Schmidt, Herman      20
## 17 994  Iwano, Masahiro      26
## 18 1110   Smith, Paul       21
## 19 1330  Onstad, Richard     19
## 20 1523 Zugnoni, Arthur A.   21
## 21 1639   Choy, Wanda       23
## 22 2398 Wallace, Maggie J.   19
## 23 4901   Bailey, Chas M.    19
## 24 5119   Bono, Sonny       24
## 25 5219  Schwarz, Jason B.   15
```

1.6

Which employees have a last name ending with ???son???

```
query = "select * from jbemployee where name like \"%son,%\";"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##      id          name salary manager birthyear startyear
## 1 26 Thompson, Bob  13000      199      1930      1970
```

1.7

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.

```
query = "select *
from jbitem
where supplier=
(select id from jbsupplier where name=\"Fisher-Price\");"
rs = dbSendQuery(mydb, query)
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 4 imported
## as numeric
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##      id          name dept price qoh supplier
## 1  43          Maze   49  325 200        89
## 2 107 The 'Feel' Book   35  225 225        89
## 3 119   Squeeze Ball   49  250 400        89
```

1.8

Formulate the same query as above, but without a subquery

```
query = "select t1.*, t2.name supplier_name from jbitem t1 join jbsupplier t2 on t1.supplier=t2.id where"
rs = dbSendQuery(mydb, query)
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 4 imported
## as numeric
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##      id          name dept price qoh supplier supplier_name
## 1  43          Maze   49  325 200        89 Fisher-Price
## 2 107 The 'Feel' Book   35  225 225        89 Fisher-Price
## 3 119   Squeeze Ball   49  250 400        89 Fisher-Price
```

1.9

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

```
query = "select * from jbcity where id in (select city from jbsupplier);"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

```
##      id      name state
## 1   10      Amherst Mass
## 2   21      Boston Mass
## 3  100      New York  NY
## 4  106      White Plains Neb
## 5  118      Hickville Okla
## 6  303      Atlanta   Ga
## 7  537      Madison   Wisc
## 8  609      Paxton    Ill
## 9  752      Dallas    Tex
## 10 802      Denver    Colo
## 11 841 Salt Lake City Utah
## 12 900      Los Angeles Calif
## 13 921      San Diego  Calif
## 14 941      San Francisco Calif
## 15 981      Seattle    Wash
```

1.10

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.)

```
query = "select name, color from jbparts where weight > (select weight from jbparts where name = \"card reader\")"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

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

1.11

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

```
query = "select t1.name, t1.color from jbparts t1 join jbparts t2 where t2.name = \"card reader\" and t1.weight > t2.weight"
rs = dbSendQuery(mydb, query)
```

```
data = fetch(rs, n=-1)
print(data)
```

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

1.12

What is the average weight of black parts?

```
query = "select avg(weight) avg_weight from jbparts where color=\"black\";"
rs = dbSendQuery(mydb, query)

## Warning in .local(conn, statement, ...): Decimal MySQL column 0 imported as
## numeric

data = fetch(rs, n=-1)
print(data)

##      avg_weight
## 1          347.25
```

1.13

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.

```
query = "select t1.supplier, t3.name, sum(t1.quan*t2.weight) total_weight from jbsupply t1 join jbparts
rs = dbSendQuery(mydb, query)

## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric

data = fetch(rs, n=-1)
print(data)

##      supplier      name total_weight
## 1          89 Fisher-Price      1135000
## 2          475      DEC          3120
```

1.14

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!

```
query = "drop table if exists jbcheapitem"
rs = dbSendQuery(mydb, query)

query = paste("CREATE TABLE jbcheapitem (",
  "id INT,",
  "name VARCHAR(20)",
  "dept INT NOT NULL,",
  "price INT,",
  "qoh INT UNSIGNED,",
  "supplier INT NOT NULL,",
  "CONSTRAINT pk_item PRIMARY KEY(id));")
rs = dbSendQuery(mydb, query)

query = "ALTER TABLE jbcheapitem ADD CONSTRAINT fk_cheapitem_dept FOREIGN KEY (dept) REFERENCES jbdept(")
```

```

rs = dbSendQuery(mydb, query)

query = "ALTER TABLE jbcheapitem ADD CONSTRAINT fk_cheapitem_supplier FOREIGN KEY (supplier) REFERENCES"

rs = dbSendQuery(mydb, query)

query = "insert into jbcheapitem (select * from jbitem where price< (select avg(price) from jbitem));"

rs = dbSendQuery(mydb, query)

query = "show tables"

rs = dbSendQuery(mydb, query)

data = fetch(rs,-1)

print(data)

```

```

##      Tables_in_andst745
## 1      jbcheapitem
## 2      jbcity
## 3      jbdebit
## 4      jbdept
## 5      jbemployee
## 6      jbitem
## 7      jbparts
## 8      jbsale
## 9      jbstore
## 10     jbsupplier
## 11     jbsupply

```

```

query = "select * from jbcheapitem"

```

```

rs = dbSendQuery(mydb, query)

```

```

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 4 imported
## as numeric

```

```

data = fetch(rs,-1)

```

```

print(data)

```

```

##      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