

# BMES 375 Homework 4 - Relational Databases

Tony K. Okeke (tko35)

## EXERCISE 1 (Students.db)

-- Create 'students' table

```
CREATE TABLE students (  
  id INTEGER PRIMARY KEY,  
  name VARCHAR(255),  
  birth DATE,  
  gpa FLOAT,  
  grad INTEGER,  
  advisor_id INTEGER  
);  
-- Add data to table  
INSERT INTO students(name, birth, gpa, grad, advisor_id)  
VALUES  
  ('Anderson', '1987-10-22', 3.9, 2009, 2),  
  ('Jones', '1990-04-16', 2.4, 2012, 1),  
  ('Hernandez', '1989-08-12', 3.1, 2011, 1),  
  ('Chen', '1990-02-04', 3.2, 2011, 1);  
-- Show table  
SELECT * FROM students;
```

	id	name	birth	gpa	grad	advisor_id
1	1	Anderson	1987-10-22	3.9	2009	2
2	2	Jones	1990-04-16	2.4	2012	1
3	3	Hernandez	1989-08-12	3.1	2011	1
4	4	Chen	1990-02-04	3.2	2011	1

-- Create 'courses' table

```
CREATE TABLE courses (  
  id INTEGER PRIMARY KEY,  
  number VARCHAR(30),  
  name VARCHAR(255),  
  quarter VARCHAR(30)  
);  
-- Add data to table  
INSERT INTO courses(number, name, quarter)  
VALUES  
  ('CS142', 'Web stuff', 'Winter 2009'),  
  ('ART101', 'Finger painting', 'Fall 2008'),  
  ('ART101', 'Finger painting', 'Winter 2009'),  
  ('PE204', 'Mud wrestling', 'Winter 2009');  
-- Show table  
SELECT * FROM courses;
```

	id	number	name	quarter
1	1	CS142	Web stuff	Winter 2009
2	2	ART101	Finger painting	Fall 2008
3	3	ART101	Finger painting	Winter 2009
4	4	PE204	Mud wrestling	Winter 2009

```
-- Create 'advisors' table
CREATE TABLE advisors (
  id INTEGER PRIMARY KEY,
  name VARCHAR(255),
  title VARCHAR(30)
);
-- Add data to table
INSERT INTO advisors(name, title)
VALUES
  ('Fujimura', 'prof'),
  ('Bolosky', 'prof');
-- Show table
SELECT * FROM advisors;
```

	id	name	title
1	1	Fujimura	prof
2	2	Bolosky	prof

```
-- Create 'courses_students' table
CREATE TABLE courses_students (
  course_id INTEGER,
  student_id INTEGER
);
-- Add data to table
INSERT INTO courses_students(course_id, student_id)
VALUES
  (1, 1), (3, 1), (4, 1), (1, 2),
  (2, 2), (1, 3), (2, 4), (4, 4);
-- Show table
SELECT * FROM courses_students;
```

	course_id	student_id
1	1	1
2	3	1
3	4	1
4	1	2
5	2	2
6	1	3
7	2	4
8	4	4

## EXERCISE 2 (Students.db):

```
-- Find all students who took 'ART101' and the quarters they took it in
SELECT s.name, c.quarter
FROM students s, courses c, courses_students cs
WHERE c.number = 'ART101' AND cs.course_id = c.id AND s.id=cs.student_id;
```

	name	quarter
1	Anderson	Winter 2009
2	Jones	Fall 2008
3	Chen	Fall 2008

### EXERCISE 3 (miRNA.db)

-- Initialize table

```
CREATE TABLE miRDB ( name VARCHAR(255), target VARCHAR(255), score FLOAT )
```

-- Import data from text file to miRDB table

```
.import 'miRDB_v6.0_prediction_result.txt' miRDB
```

-- Show first 10 records in table

```
SELECT * FROM miRDB LIMIT 10;
```

	name	target	score
1	cfa-miR-1185	XM_537211	59.3438099752
2	cfa-miR-1185	XM_536047	54.527
3	cfa-miR-1185	XM_005617022	55.1716326075
4	cfa-miR-1185	XM_014117861	57.4409058608
5	cfa-miR-1185	XM_014107884	57.1519
6	cfa-miR-1185	XM_005626419	67.0536
7	cfa-miR-1185	XM_005618203	62.64
8	cfa-miR-1185	NM_001252367	58.9597687709186
9	cfa-miR-1185	XM_005621885	78.529415636
10	cfa-miR-1185	XM_005622017	58.8579982864

-- How many miRNAs are predicted to target NM\_005166?

```
SELECT target, COUNT(*)  
FROM miRDB  
WHERE target = 'NM_005166';
```

	target	COUNT(*)
1	NM_005166	27

-- How many miRNAs are predicted to target NM\_005166 and XM\_539064?

```
SELECT COUNT(*)  
FROM miRDB  
WHERE target = 'NM_005166' OR target = 'XM_539064';
```

	COUNT(*)
1	54

-- How many predicted targets (genes) of hsa-let-7a-2-3p have a prediction score  
-- of at least 80?

```
SELECT COUNT(*)  
FROM miRDB  
WHERE name = 'hsa-let-7a-2-3p' AND score >= 80;
```

	COUNT(*)
1	529

```
-- List the miRNAs and the number of their targets that have a prediction score
-- of at least 80 and group them by name. Show only top 10 rows of the result.
```

```
SELECT name, COUNT(target)
FROM miRDB
WHERE score >= 80
GROUP BY name
ORDER BY COUNT(target) DESC
LIMIT 10;
```

	name	COUNT(target)
1	mmu-miR-7116-3p	4616
2	mmu-miR-6951-3p	4194
3	hsa-miR-3163	4080
4	gga-miR-6701-3p	3913
5	gga-miR-1786	3820
6	hsa-miR-5011-5p	3623
7	hsa-miR-3613-3p	3565
8	cfa-miR-8881	3512
9	hsa-miR-190a-3p	2980
10	cfa-miR-8843	2939

## EXERCISE 4 (weighdata.sqlite)

```
-- Write and execute SQL to get all the data for patients that don't exercise
-- (these patients have timcardio AND timeresist of zero). Show 10 such patients.
```

```
SELECT *
FROM patient
WHERE timcardio = 0 AND timeresist = 0
LIMIT 10;
```

	id	age	gender	height	initweight	calintake	jobstatus	timecardio	timeresist	sleep	steps	deltaweight
1	7	21	Female	60.6551	153.3522	1500	Active	0	0	6.7495	4.2827e+03	3.9968
2	16	39	Female	57.7803	133.4423	1500	Active	0	0	7.0986	4.4624e+03	10.0285
3	18	24	Male	69.1721	177.2989	1500	Active	0	0	7.4845	3.4461e+03	-2.5926
4	19	41	Male	77.9292	174.8174	1500	Active	0	0	8.9922	4.3349e+03	0.9817
5	28	41	Male	72.4046	196.5634	1500	Active	0	0	9.1250	3.6975e+03	-0.4721
6	33	47	Female	66.2543	161.1063	2.0790e+03	Inactive	0	0	6.1339	2.8474e+03	19.0256
7	38	36	Male	71.0037	188.0001	1500	Active	0	0	6.8328	4.2698e+03	1.0449
8	41	43	Female	60.8930	120.0183	1500	Active	0	0	7.7737	3.5255e+03	3.1415
9	43	23	Female	56.2075	147.1721	1.6190e+03	Inactive	0	0	8.8673	2.2159e+03	0.8035
10	50	22	Female	66.9509	132.7390	1500	Active	0	0	5.8651	3.4462e+03	6.6849

```
-- Write and execute SQL to count how many patients that don't exercise.
```

```
SELECT COUNT(*)
FROM patient
WHERE timcardio = 0 AND timeresist = 0;
```

	COUNT__
1	48

-- Write and execute SQL to count how many patients that are male based on gender

```
SELECT COUNT(*)  
FROM patient  
WHERE gender = 'MALE';
```

	COUNT__
1	122

-- Write and execute SQL to count how many patients are active based on jobstatus

```
SELECT COUNT(*)  
FROM patient  
WHERE jobstatus = 'Active';
```

	COUNT__
1	129