



## Welcome to Week 3

July 20–26, 2020

### Database Analyst/ Administrator

#### REMEMBER!!

Submit ALL of this week's challenges  
(or screen shots of them) to  
[experience@iechamilton.ca](mailto:experience@iechamilton.ca)  
by Sunday at noon  
for your chance to win  
**1 of 50 \$10 Gift Cards**  
or the  
**GRAND PRIZE of up to \$300 towards an  
online coding &/or technology related  
activity, camp, course or subscription**  
(subject to approval).

Have you ever thought about how companies manage their electronic information like sales or customer contacts, or wondered how searching or sorting a database works, or how hospitals share information? These are the work of Data Analysts and Administrators. Data Administrators use specialized software to store and organize data while Data Analysts inspect, clean, transform and model data with the goal of discovering useful information, which can inform a companies conclusions and support their decision-making.

Database analysts and administrators can be found in just about every career sector since all of them have information to keep. Database management can be a great career option when you wish to join a few of your many interests, providing one of them is technology! You can use your technological skills to enhance medical or sport based companies or even use it to assist not-for-profit organizations to reach their goals.

Careers in databasing may also have names such as Database Engineer or Developer. Jobs are available in both the public and private sectors and, as we continue to advance technologically, it is a rapidly growing field to enter. Salaries for this field can vary depending on where you choose to work, with the private sector and urban locations generally being a bit higher, but you can expect to enter the occupation with a yearly income of \$50,000.

There are a few courses and programs at Mohawk College that can help lead you to a career in Database Management. Take a look at some of their offerings (they'll let you know which courses you should be looking for in high school too!):

[Computer Systems Technician Software Support 548-558](#) - 2 Year Diploma Program

[Computer Systems Technology Software Development - 559](#) - 3 Year Advanced Diploma Program

CODEfest is proudly sponsored and supported by:



## Week 3

### July 20–26, 2020

# Databases

#### REMEMBER!!

Submit ALL of this week's challenges (or screen shots of them) to [experience@iechamilton.ca](mailto:experience@iechamilton.ca) by Sunday at noon for your chance to win 1 of 50 \$10 Gift Cards or the GRAND PRIZE of up to \$300 towards an online coding &/or technology related activity, camp, course or subscription (subject to approval).

## Challenge!

This week, we are taking you to Khan Academy to explore the “[Basics of SQL](#)”, a programming language for databases. You will find a series of videos to watch, instructions to read, follow and practice. To be entered to win one of the prizes this week, you will need to complete (and send in screenshots for) the following challenges/projects:

1. Book List Database
2. Box Office Hits Database
3. To Do List Database
4. Design a Store Database

## Tips:

- Be creative! You don't need to copy our examples, you can (and should) include your own ideas!
- Read the whole article (and then re-read it) for help finding how to complete each task.
- If you are struggling, Hamilton Code Clubs Camp can help! [Register](#)
- When taking screenshots, try to show as much of your coding detail as possible.
- Feel free to zoom in or send a few pictures to show both your code and the robots.

Send your completed exercises to [experience@iechamilton.ca](mailto:experience@iechamilton.ca).

Make sure you include your full name!

Prize winners will be contacted next week via information provided at registration.

You can find our finished practices on the next few pages. Be sure to be creative when completing yours!

If you are interested in exploring Databases further, here are ways to keep going!

Follow these links: [More SQL Queries](#), [Relational SQL Queries](#), [Modifying Databases](#), [Further SQL Learning](#).

CODEfest is proudly sponsored and supported by:



### REMEMBER!!

Submit ALL of this week's challenges  
(or screen shots of them) to  
[experience@iechamilton.ca](mailto:experience@iechamilton.ca)  
by Sunday at noon  
for your chance to win  
**1 of 50 \$10 Gift Cards**  
or the  
**GRAND PRIZE** of up to \$300 towards an  
online coding &/or technology related  
activity, camp, course or subscription  
(subject to approval).

### Book List Database

```

1 /** Book list:
2 Harry Potter Series (5 Stars)
3 The Inheritance Cycle (5 Stars)
4 Star Wars: Jedi Academy Trilogy (4.5 Stars)
5 **/
6
7 CREATE TABLE Books (id INTEGER PRIMARY KEY, name TEXT,
8 rating INTEGER );
9

```

**DATABASE SCHEMA**

**Books** 0 rows

id (PK)	INTEGER
name	TEXT
rating	INTEGER

Good job! You can do it!

Undo Start over

Step 1/2 Next step

```

1 /** Book list:
2 Harry Potter Series (5 Stars)
3 The Inheritance Cycle (5 Stars)
4 Star Wars: Jedi Academy Trilogy (4.5 Stars)
5 **/
6
7 CREATE TABLE Books (id INTEGER PRIMARY KEY, name TEXT,
8 rating INTEGER );
9 INSERT INTO Books VALUES(1, "Harry Potter Series", 5);
10 INSERT INTO Books VALUES(2, "The Inheritance Cycle", 5);
11 INSERT INTO Books VALUES(3, "Star Wars: Jedi Academy
12 Trilogy", 4.5);
13 SELECT * FROM Books;

```

**DATABASE SCHEMA**

**Books** 3 rows

id (PK)	INTEGER
name	TEXT
rating	INTEGER

**QUERY RESULTS**

id	name	rating
1	Harry Potter Series	5
2	The Inheritance Cycle	5
3	Star Wars: Jedi Academy Trilogy	4.5

You did it!

Undo Start over

Step 2/2 Finish challenge

### Box Office Hits Database

```

1 CREATE TABLE movies (id INTEGER PRIMARY KEY, name TEXT,
2 release_year INTEGER);
3 INSERT INTO movies VALUES (1, "Avatar", 2009);
4 INSERT INTO movies VALUES (2, "Titanic", 1997);
5 INSERT INTO movies VALUES (3, "Star Wars: Episode IV - A
6 New Hope", 1977);
7 INSERT INTO movies VALUES (4, "Shrek 2", 2004);
8 INSERT INTO movies VALUES (5, "The Lion King", 1994);
9 INSERT INTO movies VALUES (6, "Disney's Up", 2009);
10 SELECT * FROM movies;

```

**DATABASE SCHEMA**

**movies** 6 rows

id (PK)	INTEGER
name	TEXT
release_year	INTEGER

**QUERY RESULTS**

id	name	release_year
1	Avatar	2009
2	Titanic	1997
3	Star Wars: Episode IV - A New Hope	1977
4	Shrek 2	2004
5	The Lion King	1994

Getting closer! Great things are ahead!

Undo Start over

Step 1/2 Next step

CODEfest is proudly sponsored and supported by:

**REMEMBER!!**

Submit ALL of this week's challenges  
(or screen shots of them) to  
[experience@iechamilton.ca](mailto:experience@iechamilton.ca)  
by Sunday at noon  
for your chance to win  
**1 of 50 \$10 Gift Cards**  
or the  
**GRAND PRIZE** of up to \$300 towards an  
online coding &/or technology related  
activity, camp, course or subscription  
(subject to approval).

**Box Office Hits Database Cont'd.**

```

1 CREATE TABLE movies (id INTEGER PRIMARY KEY, name TEXT,
2   release_year INTEGER);
3 INSERT INTO movies VALUES (1, "Avatar", 2009);
4 INSERT INTO movies VALUES (2, "Titanic", 1997);
5 INSERT INTO movies VALUES (3, "Star Wars: Episode IV - A
6   New Hope", 1977);
7 INSERT INTO movies VALUES (4, "Shrek 2", 2004);
8 INSERT INTO movies VALUES (5, "The Lion King", 1994);
9 INSERT INTO movies VALUES (6, "Disney's Up", 2009);
10 SELECT * FROM movies;
11 SELECT * FROM movies where release_year >2000 order by
   release_year;

```

QUERY RESULTS

id	name	release_year
1	Avatar	2009
2	Titanic	1997
3	Star Wars: Episode IV - A New Hope	1977
4	Shrek 2	2004
5	The Lion King	1994
6	Disney's Up	2009

  

id	name	release_year
4	Shrek 2	Congratulations!
1	Avatar	
6	Disney's Up	

Undo Start over

Step 2/2

**To Do List Database**

```

1 CREATE TABLE todo_list (id INTEGER PRIMARY KEY, item TEXT,
2   minutes INTEGER);
3 INSERT INTO todo_list VALUES (1, "Wash the dishes", 15);
4 INSERT INTO todo_list VALUES (2, "vacuuming", 20);
5 INSERT INTO todo_list VALUES (3, "Learn some stuff on KA",
6   30);
7 INSERT INTO todo_list VALUES (4, "Submit CODEfest 2020
   Challenges", 5);

```

DATABASE SCHEMA

todo_list	4 rows
id (PK)	INTEGER
item	TEXT
minutes	INTEGER

Awesome! You can do it!

Undo Start over

Step 1/2 Next step

```

1 CREATE TABLE todo_list (id INTEGER PRIMARY KEY, item TEXT,
2   minutes INTEGER);
3 INSERT INTO todo_list VALUES (1, "Wash the dishes", 15);
4 INSERT INTO todo_list VALUES (2, "vacuuming", 20);
5 INSERT INTO todo_list VALUES (3, "Learn some stuff on KA",
6   30);
7 INSERT INTO todo_list VALUES (4, "Submit CODEfest 2020
   Challenges", 5);
8 SELECT SUM(minutes) FROM todo_list;

```

DATABASE SCHEMA

todo_list	4 rows
id (PK)	INTEGER
item	TEXT
minutes	INTEGER

QUERY RESULTS

SUM(minutes)
70

All steps complete!

Undo Start over

Step 2/2 Finish challenge

CODEfest is proudly sponsored and supported by:

#### REMEMBER!!

Submit ALL of this week's challenges  
(or screen shots of them) to  
[experience@iechamilton.ca](mailto:experience@iechamilton.ca)  
by Sunday at noon  
for your chance to win  
**1 of 50 \$10 Gift Cards**  
or the  
**GRAND PRIZE** of up to \$300 towards an  
online coding &/or technology related  
activity, camp, course or subscription  
(subject to approval).

### Design a Store Database

```

1 CREATE TABLE store (id INTEGER PRIMARY KEY, item TEXT, size TEXT,
2 price INTEGER);
3 INSERT INTO store VALUES (1, "T Shirt", "Extra Small", 15);
4 INSERT INTO store VALUES (2, "Sweater", "Extra Small", 25);
5 INSERT INTO store VALUES (3, "Hoodie", "Extra Small", 30);
6 INSERT INTO store VALUES (4, "T Shirt", "Small", 15);
7 INSERT INTO store VALUES (5, "Sweater", "Small", 25);
8 INSERT INTO store VALUES (6, "Hoodie", "Small", 30);
9 INSERT INTO store VALUES (7, "T Shirt", "Medium", 15);
10 INSERT INTO store VALUES (8, "Sweater", "Medium", 25);
11 INSERT INTO store VALUES (9, "Hoodie", "Medium", 30);
12 INSERT INTO store VALUES (10, "T Shirt", "Large", 15);
13 INSERT INTO store VALUES (11, "Sweater", "Large", 25);
14 INSERT INTO store VALUES (12, "Hoodie", "Large", 30);
15 INSERT INTO store VALUES (13, "T Shirt", "Extra Large", 15);
16 INSERT INTO store VALUES (14, "Sweater", "Extra Large", 25);
17 INSERT INTO store VALUES (15, "Hoodie", "Extra Large", 30);
18 SELECT * FROM store;

```

QUERY RESULTS

id	item	size	price
1	T Shirt	Extra Small	15
2	Sweater	Extra Small	25
3	Hoodie	Extra Small	30
4	T Shirt	Small	15
5	Sweater	Small	25
6	Hoodie	Small	30
7	T Shirt	Medium	15
8	Sweater	Medium	25
9	Hoodie	Medium	30
10	T Shirt	Large	15
11	Sweater	Large	25
12	Hoodie	Large	30

Request help

Spin-off

Review a project

```

1 CREATE TABLE store (id INTEGER PRIMARY KEY, item TEXT, size TEXT,
2 price INTEGER);
3 INSERT INTO store VALUES (1, "T Shirt", "Extra Small", 15);
4 INSERT INTO store VALUES (2, "Sweater", "Extra Small", 25);
5 INSERT INTO store VALUES (3, "Hoodie", "Extra Small", 30);
6 INSERT INTO store VALUES (4, "T Shirt", "Small", 15);
7 INSERT INTO store VALUES (5, "Sweater", "Small", 25);
8 INSERT INTO store VALUES (6, "Hoodie", "Small", 30);
9 INSERT INTO store VALUES (7, "T Shirt", "Medium", 15);
10 INSERT INTO store VALUES (8, "Sweater", "Medium", 25);
11 INSERT INTO store VALUES (9, "Hoodie", "Medium", 30);
12 INSERT INTO store VALUES (10, "T Shirt", "Large", 15);
13 INSERT INTO store VALUES (11, "Sweater", "Large", 25);
14 INSERT INTO store VALUES (12, "Hoodie", "Large", 30);
15 INSERT INTO store VALUES (13, "T Shirt", "Extra Large", 15);
16 INSERT INTO store VALUES (14, "Sweater", "Extra Large", 25);
17 INSERT INTO store VALUES (15, "Hoodie", "Extra Large", 30);
18 SELECT * FROM store;
19 SELECT * FROM store order by price;
20

```

id	item	size	price
1	T Shirt	Extra Small	15
4	T Shirt	Small	15
7	T Shirt	Medium	15
10	T Shirt	Large	15
13	T Shirt	Extra Large	15
2	Sweater	Extra Small	25
5	Sweater	Small	25
8	Sweater	Medium	25
11	Sweater	Large	25
14	Sweater	Extra Large	25
3	Hoodie	Extra Small	30
6	Hoodie	Small	30
9	Hoodie	Medium	30
12	Hoodie	Large	30
15	Hoodie	Extra Large	30

Start over

Request help

Save

Don't forget to check out more SQL practices below!

Follow these links:  
[More SQL Queries](#)  
[Relational SQL Queries](#)  
[Modifying Databases](#)  
[Further SQL Learning](#).

CODEfest is proudly sponsored and supported by: