Introduction to Databases

Author: Ernest Edinboro

Coauthor: Caroline Skalla

Date: Summer 2018

Question	Answers
What type of database type does Shodor use?	**Relational Database**
What DBMS does Shodor use?	**mySQL**

MySQL Building My First Table

SQL	MySQL Syntax	Results
Building Table and Table Info	**Write proper SQL syntax to query the People table using Sequel Pro.**	**What Was the Result**

CREATE TABLE
Step1:
	`CREATE TABLE People (`
	`  id int(11) unsigned NOT NULL auto_increment,`
	`  firstName varchar(255) default NULL,`
	`  lastName varchar(255) default NULL,`
	
	`  favColor varchar(255) default NULL,`
	`  shoeSize decimal (3,1)default NULL,`
	`  PRIMARY KEY  (id)`
	`) ENGINE=InnoDB DEFAULT CHARSET=utf8;`
		Created a table with the given data points in it.

INSERT INTO
Step2:
	`INSERT INTO People (firstName, lastName,favColor, shoeSize)`
	`VALUES`
	`('Ernie','Edinboro','Orange',11.5),`
	`('Gustavo','Nunez-Vivanco','Blue',9.5),`
	`('Chris','Parker', 'Red', 10),`
	`('Huny','Chung','Light Green', 9),`
	`('Aaron','Weeden', 'Purple', 10),`
	`('No','Hope', 'Black', 4),`
	`('Shawn','Kang', 'Red', 10),`
	`('Divya','Aikat','Purple',9.5),`
	`('Michael','Allen Dacany', 'Blue', 12),`
	`('Naylea ','Jacobo', 'Blue', 7.5),`
	`('Mihir','Khadri', 'Red', 10.5),`
	`('Krista', 'Katzenmeyer', 'Cupcake', 8.5),`
	`('Erica','Aiello', 'Yellow', 3),`
	`('Joel','Coldren', 'Green', 13.5),`
	`('Rohin','Shahi', 'Red', 12.5),`
	`('Ameya','Rao', 'Blue', 9.5),`
	`('Luka','Ashe-Jones', 'Orange', 8),`
	`('Ernie','Edinboro', 'Blue', 12),`
	`('Ernie','Edinboro', 'Green', 11),`
	`('Leanne','Chan', 'Green',6),`
	`('Erica','Aiello', 'Purple', 3.5);`
		Inserts more data into the table.

ALTER Table **Step3:**	`ALTER TABLE People` `ADD coinFlip VARCHAR(5) NOT NULL;`	**Changes a data point in the table**

Add Data to the newly created coinFlip field.
Step4:
	`INSERT INTO People (id, coinFlip)`
	`VALUES`
	`('1','Tails'),('2','Tails'),('3','Tails'),`
	`('4','Heads'),('5','Heads'),('6','Tails'),`
	`('7','Heads'),('8','Tails'),('9','Heads'),`
	`('10','Heads'),('11','Tails'),('12','Heads'),`
	`('13','Heads'),('14','Heads'),('15','Tails'),`
	`('16','Heads'),('17','Heads'),('18','Heads'),`
	`('19','Heads'),('20','Tails'),('21','Heads')`
	`ON DUPLICATE KEY UPDATE`
	`id=VALUES(id), coinFlip=VALUES(coinFlip);`
		Added new data to the table

MySQL Syntax Guide:
Day 1

SQL	MySQL Syntax	Results
Other usefully SQL for Building Table and Table Info	Write proper SQL syntax to query the People table using Sequel Pro.	What Was the Result
SHOW TABLEs	`SHOW tables;`	Returns the tables that are in the database.
DESCRIBE TABLE	`DESCRIBE People;`	Displays information about each of the table's columns.
DROP TABLE Don't run unless you really want to get rid of a table and all its data. You will have to start over. DO NOT RUN QUERY!	`DROP TABLE People;`	DO NOT RUN this query unless you want to destroy a table and data. Once done it cannot be undone.

NOT NULL	`SELECT *`
	`FROM People`
	`WHERE firstName IS NOT NULL;`
		Returns values of table that are not null.

PRIMARY KEY	`PRIMARY KEY  (id)`	Creates a unique id for each row in the table
AUTO_INCREMENT	`id int(11) unsigned NOT NULL auto_increment,`	Increments the id every time a new one is created

DELETE **Be careful! DO NOT RUN QUERY!**	`DELETE FROM People;`	**Do not run query**
Useful SQL Syntax		

UPDATE	`UPDATE People`
	`SET shoeSize = 5,`
	`WHERE coinFlip= 'Tails';`
		Changes a value

SELECT	`SELECT *` `FROM People;`	**Returns data from a specified table or location in a table.**

WHERE	`Select *`
	`FROM People`
	`WHERE shoeSize = 8;`
		Specify a selection by location.

ORDER BY	`SELECT *`
	`FROM People`
	`ORDER BY shoeSize;`
		Sorts in ascending order

AND, OR, Not	`SELECT *`
	`FROM People`
	`WHERE shoeSize = 10 OR shoeSize = 8;`
		Selects data that and, or, not meet the conditions.

Count, Avg, Sum	`SElECT COUNT(shoeSize)`
	`FROM People;`
	
	`SELECT AVG(shoeSize)`
	`FROM People;`
	
	`SELECT SUM(shoeSize)`
	`FROM People;`
		Returns number of data entries in that column

LIKE	`SELECT firstName`
	`FROM People`
	`WHERE firstName LIKE 'a%';`
		Searches for a specified pattern in a column

IN	`SELECT firstName, lastName, shoeSize`
	`FROM People`
	`WHERE shoeSize IN (5, 8, 9, 10);`
		Finds a match

BETWEEN	`SELECT shoeSize`
	`FROM People`
	`WHERE shoeSize BETWEEN 0 AND 5;`
		Finds data between two values

ALIAS	`SELECT favColor AS color` `FROM People;`	**Gives a temporary name.**
SQL Operators	**Best to use Arithmetic and Comparison Operators on shoeSize field**	
Arithmetic Operators		
Add	`SELECT shoeSize + 1` `FROM People;`	`addition`
Subtraction	`SELECT shoeSize - 2` `FROM People;`	`subtracts`
Multiplication	`SELECT shoeSize * 3` `FROM People;`	`multiplies`
Division	`SELECT shoeSize / 4` `FROM People;`	`divides`
Modulos	`SELECT shoeSize % 2`	`Finds remainder`

	`FROM People;`	
Comparison Operators		

Equal to	`SELECT *`
	`FROM People`
	`WHERE lastName = 'Edinboro';`
		Returns data values that meet condition equal to the specified value.

Greater than >	`SELECT *`
	`FROM People`
	`WHERE shoeSize > 6;`
		Returns data values that meet condition greater than specified value.

Less than <	`SELECT *`
	`FROM People`
	`WHERE shoeSize < 9;`
		Returns data values that meet condition less than specified value.

Greater than equal to >=	`SELECT *`
	`FROM People`
	`WHERE shoeSize >= 6;`
		Returns data values that meet condition greater than or equal to the specified value.

Less than equal to <=	`SELECT *`
	`FROM People`
	`WHERE shoeSize <= 9;`
		Returns data values that meet condition less than or equal to the specified value.

Not equal to <>	`SELECT *`
	`FROM People`
	`WHERE shoeSize <> 10;`
		Returns data values that meet condition not equal to the specified value.

SQL Data Types

List 10 SQL Data Types **You cannot use Int or VARCHAR**	**What types of Data do they Store?**	
Boolean	`True or false.`	

SMALLINT	**Integer with precision 5.**	
Float	**Number, with precision 16.**	
Array	**Collection of elements of a set length, and is ordered.**	
Time	**Stores hour, minute, and second values.**	
XML	**Stores XML data.**	
DATE	**Stores year, month, and day values.**	
BIGINT	**Stores integer with 19 precision.**	
MULTISET	**An unordered collection of elements that can be variable length.**	
BINARY	**A binary string with a fixed length.**	

Writing SQL queries

SQL MINI Challenge **Cannot use INT or VARCHAR**	**Write SQL query to answer the questions**	**Result**

What is Erica's favorite color?	`SELECT favColor`
	`FROM People`
	`WHERE firstName = 'Erica';`
		`Yellow and Purple`

Whose shoe size is 3?
Show First and last name
	`SELECT firstName, lastName`
	`FROM People`
	`WHERE shoeSize = 3;`
		`Erica Aiello`

Whose favorite color is Cupcake?
Show First Name only
	`SELECT firstName`
	`FROM People`
	`WHERE favColor ='Cupcake';`
		`Krista`

What is the average shoe size?	`SELECT AVG(shoeSize)` `FROM People;`	`9.18182`

Whose favorite color is Red? Show Last Name only ORDER BY ASCending order.	`SELECT lastName`
	`FROM People`
	`WHERE favColor ='red'`
	`ORDER BY lastName;`
		`Kang`
		`Khadri`
		`Parker`
		`Shahi`

What is the favorite color of people with shoe size 10?	`SELECT favColor`
	`FROM People`
	`Where shoeSize = 10;`
		`Red and Purple`

Select all data for user(s) id =13	`SELECT *`
	`FROM People`
	`Where id = 13;`
		`13  Erica Aiello`
		`Yellow3.0  Heads`

Select first name for user(s) where favorite color is red	`SELECT firstName`
	`FROM People`
	`WHERE favColor ='red';`
		`Chris`
		`Shawn`
		`Mihir`
		`Rohin`

Select favorite color where shoe size greater than or equal to 12	`SELECT favColor`
	`FROM People`
	`WHERE shoeSize >= 12;`
		`Blue`
		`Green`
		`Red`
		`Blue`

Select shoe size where first name = Joel	`SELECT shoeSize`
	`FROM People`
	`WHERE firstName = 'Joel';`
		`13.5`

Select lastname where favorite color is black	`SELECT lastName`
	`FROM People`
	`WHERE favColor = 'black';`
		`Hope`

Insert into People Keith,Kelly, Black, 11, Tails	`INSERT INTO People (firstName, lastName, favColor, shoeSize, coinFlip)`
	`VALUES`
	`('Keith', 'Kelly', 'Black', 11, 'Tails')`
		`22  Keith Kelly Black 11.0  Tails`

	ON DUPLICATE KEY UPDATE id=VALUES(id), coinFlip=VALUES(coinFlip);	
Update First name of id 1 to Ernest	UPDATE People SET firstName = "Ernie" WHERE id = 1;	1 Ernie Edinboro Orange11.5 Tails

MySQL Syntax Guide:
Day 2

Database Interaction	MySQL Syntax using People table	What were the results?
PRIMARY KEY	PRIMARY KEY (id)	Makes the primary key called id.
FOREIGN KEY		Makes another key that can relate to a column.
JOIN		
INNER JOIN		

LEFT JOIN		
RIGHT JOIN		
GROUP BY		

SQL Snippet for creating Relational DB	`CONSTRAINT CoinFlip_ibfk_1`
	`FOREIGN KEY (peopleId) REFERENCES People (id)`
	`ON DELETE CASCADE ON UPDATE CASCADE,`	

SQL MINI Challenge	**SQL Syntax used**	**Result**
?		
?		
?		
?		

MySQL Syntax Guide:
Day 2

Database Interaction	MySQL Syntax using newly created relational database People, coinFlips, color, table, shoes	What were the results?
PRIMARY KEY	`id int(11) unsigned NOT NULL auto_increment,`	`Automatically creates a primary key`
FOREIGN KEY	`peopleID int(11) unsigned NOT NULL,`	`Creates a foreign key`

JOIN	SELECT cfPeople.lName, cfCoinFlip.results FROM cfPeople JOIN cfCoinFlip ON cfPeople.id=cfCoinFlip.peopleID;	**T heads** **T tails** **T tails** **K tails** **K heads** **W heads** **W tails** **W heads** **H heads** **H heads** **H tails** **H heads** **H heads** **E heads** **E heads** **E tails**
INNER JOIN	SELECT cfPeople.lName, cfCoinFlip.results FROM cfPeople INNER JOIN cfCoinFlip ON cfPeople.id=cfCoinFlip.peopleID;	**T heads** **T tails** **T tails** **K tails** **K heads** **W heads** **W tails** **W heads** **H heads** **H heads** **H tails** **H heads** **H heads** **E heads** **E heads** **E tails**
LEFT JOIN	SELECT cfPeople.lName, cfCoinFlip.results FROM cfPeople LEFT JOIN cfCoinFlip ON cfPeople.id=cfCoinFlip.peopleID;	**T heads** **T tails** **T tails** **K tails** **K heads** **W heads** **W tails** **W heads**

		H heads H heads H tails H heads H heads E heads E heads E tails S NULL
RIGHT JOIN	SELECT cfColor.colorChoice, cfCoinFlip.results FROM cfCoinFlip LEFT JOIN cfColor ON cfColor.peopleID=cfCoinFlip.peopleID;	green heads green tails green tails blue tails blue heads red heads red tails red heads orangeheads orangeheads orangetails orangeheads orangeheads purpleheads purpleheads purpletails NULL tails
GROUP BY	**SELECT cfCoinFlip.results, COUNT(*) FROM cfCoinFlip GROUP BY cfCoinFlip.results;**	heads 10 tails 7
SQL Snippet for creating Relational DB	**CONSTRAINT CoinFlip_ibfk_1 FOREIGN KEY (peopleId) REFERENCES People (id) ON DELETE CASCADE ON UPDATE CASCADE,**	
